

Lawrence Livermore National Laboratory
UCRL-52504

HCT
A General Computer Program
for Calculating Time-Dependent Phenomena
Involving One-Dimensional
Hydrodynamics, Transport,
and Detailed Chemical Kinetics

Carl M. Lund

Originated: August 2, 1978

Revised: August 8, 1995
By: Lila Chase

For: Charles K. Westbrook
and William J. Pitz

Abstract

We describe a general computer program called HCT that is designed to calculate time-dependent problems involving one-dimensional gas hydrodynamics, transport, and detailed chemical kinetics. HCT is designed to handle a wide variety of calculations in the framework of a general “robust” integrator. We describe in detail the difference equations used and the reasoning governing their selection. The solution of the difference equation by a generalized Newton’s iteration scheme is described. The structure of the coding as implemented on the CDC 7600 ¹ at Lawrence Livermore Laboratory is explained. An appendix gives a glossary of the variables needed to run problems. A complete listing of the program is given in a second appendix. Several sample problems are described to illustrate the behavior and capabilities of HCT. A comparison with other programs that calculate similar phenomena is given.

¹In the past, HCT has been implemented on Cray-1s, Cray-X/MPs Cray Y/MPs running LTSS and NLTSS at LCC, Cray-1s and Cray-X/MPs running CTSS at NERSC (then MFECC), Crays running UNICOS, Sun 3/50s (OS4.0), Sun 4/260s, MIPS-120, Sparc Solaris, VAX VMS, and PC 586s running Windows and Microsoft FORTRAN Powerstation. It currently runs primarily on IBM RS/6000s running AIX and Sun Sparcs running Solaris.

Contents

1	Introduction	1
2	Difference Equations	4
3	Differential Equations	11
4	Solution Technique	15
5	Eulerian Calculations	24
6	Dynamic Rezoning	29
7	Kinetics Calculations	37
8	Boundary Calculations	40
9	Energy Sources	43
10	Calculational Cycle	46
11	Some Programming Details	50
12	Examples	54
13	Relation To Other Methods	70
14	Acknowledgements	76

A	Sample Output	80
B	Sample NAMELIST Input for Various Types of Problems	116
B.1	CH4-72	116
B.2	SANT89	120
C	Definition of NAMELIST Variables	130

List of Figures

A.1	Test problem involving only hydrodynamics and thermal conduction after initial cycle.	80
A.2	Test problem involving only hydrodynamics and thermal conduction after 50 cycles.	81
A.3	Ozone test problem after initial cycle.	84
A.4	Ozone test problem after 100 cycles.	85
A.5	Ozone test problem after flame has propagated past the middle of the calculational grid.	86
A.6	Velocity of flame relative to the unburned gas in ozone test problem.	90
A.7	Ozone test problem just after steady-state propagating flame has been established.	91
A.8	Comparison of calculated flame velocities as a function of time for 25-zone ozone test problem (curve {1}) with velocities from calculations of the same problem run with 35 zones (curve {2}) and 50 zones (curve {3}).	92
A.9	Comparison of calculated temperature profile at $400\mu sec$ for 25-zone ozone test problem (curve {1}) with profiles from calculations of the same problem using 35 zones (curve {2}) and 50 zones (curve {3}).	93
A.10	Comparison of calculated flame velocities as a function of time for the 25-zone ozone test problem (curve {1}) with calculations of the same problem with varying running parameters. Curve {2} represents a calculation with reduced timesteps, curve {3} represents a calculation with more zones in the reaction region, and curve {4} represents a calculation with more total zones.	94
A.11	Methane test problem after initial cycle.	100

A.12 Left tenth of grid of the methane test problem at early stages of initiation.	101
A.13 Left tenth of grid of the methane test problem as flame begins to propagate away from source region.	102
A.14 Methane test problem after propagating flame has moved across most of the grid.	103

List of Tables

A.1	Input deck used to generate ozone test problem.	82
A.2	Data associated with the equations of state used in the ozone test problem.	83
A.3	Data associated with the reaction set used in the ozone test problem.	83
A.4	General Lagrange edit of ozone test problem at cycle 1000. .	87
A.5	Lagrange edit of rates affecting the physical variables in zone 12 for ozone test problem at cycle 1000.	88
A.6	Lagrange edit of rates affecting the physical variables in zone 4 for ozone test problem at cycle 1000.	89
A.7	Data associated with the equations of state used in the methane test problem.	95
A.8	Data associated with the reaction set used in the methane test problem.	96
A.8	Data associated with the reaction set used in the methane test problem: continued.	97
A.9	Input deck used to generate methane test problem.	98
A.9	Input deck used to generate methane test problem: continued.	99
A.10	General Lagrange edit of methane test problem at cycle 1500.	104
A.11	Edit of species mole fractions in methane test problem at cycle 1500.	105
A.11	Edit of species mole fractions in methane test problem at cycle 1500: continued.	106
A.12	Lagrange edit of rates affecting physical variables in zone 13 for methane test problem at cycle 1500.	107

A.12 Lagrange edit of rates affecting physical variables in zone 13 for methane test problem at cycle 1500: continued.	108
A.12 Lagrange edit of rates affecting physical variables in zone 13 for methane test problem at cycle 1500: continued.	109
A.13 Edit of the relative changes in the physical variables during the Lagrange part of cycle 1500 of the methane test problem.	110
A.13 Edit of the relative changes in the physical variables during the Lagrange part of cycle 1500 of the methane test problem: continued.	111
A.14 Edit of the relative changes in the physical variables due to the remap onto the moving grid of the methane test problem at cycle 1500.	112
A.14 Edit of the relative changes in the physical variables due to the remap onto the moving grid of the methane test problem at cycle 1500: continued.	113
A.15 Edit of the relative changes in the physical variables due to the remap onto the cycle 1500 grid of the methane test problem at cycle 1501.	114
A.15 Edit of the relative changes in the physical variables due to the remap onto the cycle 1500 grid of the methane test problem at cycle 1501: continued.	115

Introduction

In this paper we describe a computer code HCT (Hydrodynamics, Chemistry, and Transport) that is capable of modeling in detail one-dimensional time-dependent combustion phenomena of gases. The physical processes that are modeled are chemical reactions, thermal conduction, species diffusion, and hydrodynamics.

HCT is currently implemented on the CDC 7600 at Lawrence Livermore Laboratory. On this machine, a typical practical problem might involve 20 chemical species and 50 elementary reactions with rates of the Arrhenius form. Such a time-dependent calculation would be expected to require 10 minutes of computer time. Several sample problems are described in this report to illustrate the capabilities of the code.

There are several considerations that govern the most effective way of writing down difference equations to describe such problems. The first is that the concentrations of species change on a timescale long compared to sound transit times across the grid. Physically, this is because chemical reactions proceed at a rate that is slow compared to the rate of collisions in the gas. Similarly, thermal conduction is slow compared to sound transit times. These considerations make it advantageous to look at implicitly differenced hydrodynamics equations. Implicitly differenced equations have the property that they give the equilibrium configuration solution when a timestep Δt is used that is long compared to sound transit times. They are thus stable when the Courant condition, which limits the timestep to less than the sound transit time across any zone, is violated. This is not true in general for time-centered or explicit hydro difference equations. Of course, if Δt is large compared to the Courant Δt , short timescale phenomena are suppressed. For example, it is clear that one is not following sound signals across the grid. However, these phenomena are not governing the evolution of the system or else the species concentrations would be changing on the shorter timescale.

A second factor that influences the selection of difference equations is the nature of chemical kinetics. These systems are often “stiff”. Stiffness refers to the range of timescales in the problem compared to the timescales

of evolution of the system. In chemical systems one often has subsets of the system that are in equilibrium in such a way that any variations from equilibrium will decay to equilibrium on a timescale very short compared to the overall variations of the system. Again, if we difference the chemical kinetics equations implicitly and use large timesteps, the solution will allow these subsets to remain in equilibrium without having the solution become numerically unstable.

Another physical factor affecting the numerical methods we chose is the close coupling between the chemistry in one part of the fluid and its effect on its neighbors. Subsonic deflagration phenomena, such as flames, proceed in a medium when the reactions at one point transfer a large amount of energy and species to their neighboring points. Accuracy depends on a careful calculation of this interaction. Thus for a given timestep, a numerical scheme that has the chemistry and transport terms differenced at the same time can be expected to be more accurate than one in which the transport and chemistry are solved separately and then combined.

Finally, we have written HCT to allow a description of flame phenomena even though the length scale of the problem may vary considerably across the grid. We have included a subroutine that determines a mesh with high resolution in regions of high temperature gradients. Since the chemistry is strongly affected by temperature, high temperature gradients usually coincide with high gradients in the other physical quantities as well. The code can then continuously update the mesh to maintain the optimum zoning as the problem is running.

This is accomplished by writing the difference equations in the Lagrange formulation and allowing a remap of the physical variables onto another grid. It is particularly easy to write down stable Lagrangian hydro difference equations, since they describe the system on a grid that moves with the fluid and thus have no advection terms. Adding a remapping algorithm allows both Eulerian and Lagrangian calculations to be done with the same code.

If the remap routine is not called, a problem is calculated in a pure Lagrangian manner. A pure Lagrange calculation may be the best way to run a problem if one wishes to minimize the effects of numerical diffusion, or if the problem does not have large changes in fluid configurations.

If one desires, the remap can be chosen so that after each Lagrange cycle the problem configuration is mapped onto a fixed grid. This is equivalent to the usual Eulerian calculation with a fixed grid.

If one calls a separate routine to determine a new grid before the remap is done, one can put the resolution where it is most needed. One then has an Eulerian calculation with a variable grid.

Separation of the remap from the rest of the calculation also has the advantage of simplifying the coding of different remap possibilities. We allow one remap that is first order in spatial separations, and two remap schemes that are second order in separations.

The main emphasis in the code's development has been on general algorithms that are as "robust" as possible. Thus while it may be possible to design algorithms that are faster for a particular problem, it is hoped that HCT will reliably calculate a wide variety of problems in a reasonable amount of computer time with little or no special "tuning" for any particular problem.

Difference Equations

In this section we write down the difference equations solved in the Lagrange part of the calculational cycle. The manner in which the difference equations reduce to the appropriate differential equations is shown in the following section.

The difference equations are derived by considering the problem to be divided into small cells, or zones. Writing down the conservation laws for species and energy in these cells, and the equation of motion for the cell boundaries, gives the difference equations for the system. This is basically the control volume approach for deriving finite difference equations.[1]

The cells are described by their thickness Δr_j . This variable has units of centimetres. The velocity of the boundary between zone j and $j - 1$ is denoted $v_{j-\frac{1}{2}}$. Its units are *centimetres/second*. The concentration of species i in zone j is denoted by $c_{i,j}$. Units of *moles/cubic centimetre* are used.

The $c_{i,j}$ species are assumed to be ideal gases at temperature T_j . It is assumed that the same T_j applies for all species $c_{i,j}$ in zone j . Temperature has units of *kelvins*.

The state of the system at time t is completely specified if the quantities $c_{i,j}$, T_j , Δr_j and $v_{j-\frac{1}{2}}$ are known for all i, j . We then write difference equations that determine the evolution of these quantities from time t to time $t + \Delta t$.

We use the notation that unprimed quantities denote quantities evaluated at time t , the beginning of a cycle. Prime quantities are evaluated at the end of the cycle, time $t + \Delta t$. The method of solution of these equations is explained in a following section.

Conservation of species gives the difference equation

$$c'_{i,j} - \frac{\Delta r_j}{\Delta r'_j} c_{i,j} = \sum_k \Delta t R'_{i,j,k} - \frac{\Delta t}{\Delta r'_j} \left(F^c_{i,j+\frac{1}{2}} - F^c_{i,j-\frac{1}{2}} \right) \quad (2.1)$$

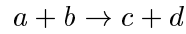
$R'_{i,j,k}$ is the contribution of the k^{th} reaction to the change of the i^{th} species. $F^c_{i,j-\frac{1}{2}}'$ is the diffusion flux of species i through the $j - \frac{1}{2}$ boundary.

The $R_{i,k}$ term consists of a temperature-dependent Arrhenius term,

$$f_k(T) = A_k T^{b_k} \exp(-E_k^a/RT), \quad (2.2)$$

times the appropriate species concentrations going into the reaction. (We have dropped the zone index j in Eq. (2.2)). R is the univesal gas constant. E^a is an activation energy. The code stores E^a in *calories/mole*.

For example, the reaction



has a rate \mathcal{R}_k given by

$$\mathcal{R}_k = f_k(T) c_a c_b, \quad (2.3)$$

and gives rise to the contributions

$$R_{a,k} = R_{b,k} = -R_{c,k} = -R_{d,k} = -\mathcal{R}_k.$$

The diffusion flux is given by

$$F^c_{i,j-\frac{1}{2}}' = -(\rho D_i)_{j-\frac{1}{2}} \frac{1}{\Delta r'_{j-\frac{1}{2}}} \left(\frac{c'_{i,j}}{\rho'_j} - \frac{c'_{i,j-1}}{\rho'_{j-1}} \right) \quad (2.4)$$

If the D_i 's are the same for all species, Eq. (2.4) has the desirable property that no mass diffuses between zones. This is seen from multiplying Eq. (2.1) by the atomic weight A_i and summing over species.

The boundary-centered $\Delta r'_{j-\frac{1}{2}}$ is given by

$$\Delta r'_{j-\frac{1}{2}} = \frac{1}{2} [\Delta r'_{j-1} + \Delta r'_j] \quad (2.5)$$

For simplicity, the $(\rho D_i)_{j-\frac{1}{2}}$ term is evaluated at the beginning of the cycle, but the other terms are evaluated at the end. The code calculates ρD_i at the center of each zone and then gets boundary values by averaging:

$$(\rho D_i)_{j-\frac{1}{2}} = \frac{1}{2} [(\rho D_i)_{j-1} + (\rho D_i)_j] \quad (2.6)$$

D_i is allowed to have a power-law dependence on the total species concentration c_{tot} and T :

$$D_i = D_i^0 T^\alpha c_{tot}^\beta. \quad (2.7)$$

Notice that all the terms on the right side of Eq. (2.1) are expressed as functions of the variables at the end of the cycle. This choice is forced on us because of the stiffness of the kinetics terms in the problems of interest. This stiffness is illustrated by considering the contributions $R_{i,k}$ of Eq. (2.3) as defining a characteristic time $\tau_{i,k}$ through

$$R_{i,k} = f_k(T) c_l c_m \equiv \frac{c_i}{\tau_{i,k}}$$

$\tau_{i,k}$ is the timescale of variation in c_i due to reaction k alone.

Neglecting diffusion and assuming $\Delta r_j / \Delta r'_j = 1$, Eq. (2.1) becomes

$$c'_i - c_i = c'_i \sum_k \frac{\Delta t}{\tau'_{i,k}} \quad (2.8)$$

The problem is stiff when there exists $\tau_{i,k}$ such that $\Delta t / \tau_{i,k} \gg 1$. In this case the left side of Eq. (2.8) is negligible compared to the right side, so that c'_i is determined by the condition

$$0 = \sum_k R'_{i,k}$$

That is, c_i is in kinetic equilibrium, as we want.

If $\tau_{i,k}$ were evaluated using c_j instead of c'_j , the only term containing c'_j in Eq. (2.8) is on the left side. Slight mismatches in the large (but nearly cancelling) terms contributing to the right side would result in very large errors in c'_i . The calculation then quickly becomes numerically unstable.

To get the rate of change of the boundary velocities, we write down Newton's second law as

$$m_{j-\frac{1}{2}} \left(v'_{j-\frac{1}{2}} - v_{j-\frac{1}{2}} \right) = -\Delta t \left[(P+Q)'_{j+1} - (P+Q)'_j \right] \quad (2.9)$$

$m_{j-\frac{1}{2}}$ is the mass associated with the $j - \frac{1}{2}$ boundary. We take this to be the average of the masses in zone j and $j - 1$,

$$m_{j-\frac{1}{2}} = \frac{1}{2} \left[\Delta r_{j-1} \sum_i A_i c_{i,j-1} + \Delta r_j \sum_i A_i c_{i,j} \right] \quad (2.10)$$

where A_i is the atomic weight of species i .

The pressure P_j in the j^{th} cell is given by the ideal gas law as

$$P_j = \sum_i c_{i,j} RT. \quad (2.11)$$

The code carries pressure in CGS units. Q is a viscosity, which may either be real or artificial. We have not in general found it necessary to specify a Q to give stability as long as shocks are not present. The code allows the form

$$Q_j = -\lambda \frac{1}{\Delta r_j} (v_{j+\frac{1}{2}} - v_{j-\frac{1}{2}}). \quad (2.12)$$

It is important to note that $P + Q$ is evaluated at the cycle end time. As Δt becomes large, the left side of Eq. (2.9) becomes negligible, and the solution to Eq. (2.9) satisfies

$$P'_{j-1} = P'_j.$$

The code then gives the equilibrium solution at the end of the cycle.

A general statement about this implicit formulation of the hydrodynamics equations is that when Δt is greater than the Courant Δt , it tends to move the fluid toward pressure equilibrium. Then the velocity of the fluid is not determined so much by the acceleration equation as by how far the fluid elements must move in time Δt so that the system comes to pressure equilibrium.

We can estimate when this condition is satisfied by noting that P is on the order of ρc^2 , where c is the sound speed in the medium. Then the ratio of the left side of Eq. (2.9) to the right side is of order

$$\frac{\Delta r}{c \Delta t} \frac{v}{c}$$

The first factor is small when the sound transit time across the zone is short compared to Δt . The second factor is small when the velocity is small compared to c .

The equation for the energy in the zone gives

$$\left[\Delta r'_j E'_j - \Delta r_j E_j \right] + (P + Q)'_j \left[\Delta r'_j - \Delta r_j \right] = -\Delta t \left[F_{j+\frac{1}{2}}^{E'} - F_{j-\frac{1}{2}}^{E'} \right] \quad (2.13)$$

E'_j is the internal energy density in the zone, given by

$$E_j = \sum_i c_{i,j} \epsilon_i \quad (2.14)$$

where ϵ_i is the internal energy density of species i . Note that ϵ_i has units of *ergs/mole*, while E_j has units of *ergs/cm³*. The code uses fifth-order polynomial fits in T to the JANAF data[2] for the specific heat at constant pressure. The internal energy is then given by

$$\epsilon_i = \Delta H_0^i + \int_{T_0} c_P^i dT - RT \quad (2.15)$$

where ΔH_0^i is the enthalpy at T_0 .

The second term in Eq. (2.13) represents the $(P + Q)dV$ work done in the cycle. This term is differenced implicitly since the corresponding term in Eq. (2.9) is differenced implicitly. The right side represents energy lost by diffusion. There are two contributions to it, one represents the enthalpy carried by the species diffusion, the second is the thermal conduction contribution. Thus

$$F_{j-\frac{1}{2}}^E{}' = F_{j-\frac{1}{2}}^T{}' + F_{j-\frac{1}{2}}^H{}', \quad (2.16)$$

where

$$F_{j-\frac{1}{2}}^T{}' = - \left(\alpha^T C_P \right)_{j-\frac{1}{2}} \frac{1}{\Delta r'_{j-\frac{1}{2}}} \left(T_j' - T_{j-1}' \right) \quad (2.17)$$

$$\left(\alpha^T C_P \right)_{j-\frac{1}{2}} = \frac{1}{2} \left[\left(\alpha^T C_P \right)_{j-1} + \left(\alpha^T C_P \right)_j \right] \quad (2.18)$$

$$C_P = \sum_i c_i c_P^i, \quad (2.19)$$

$$F_{j-\frac{1}{2}}^H{}' = \sum_i h'_{i,j-\frac{1}{2}} F_{i,j-\frac{1}{2}}^c{}' \quad (2.20)$$

$$h'_{i,j-\frac{1}{2}} = \frac{1}{2} \left(h'_{i,j-1} + h'_{i,j} \right) \quad (2.21)$$

As with $F_{i,j-\frac{1}{2}}^c$, $\Delta r'_{j-\frac{1}{2}}$ is given by Eq. (2.5). Also, the diffusion coefficients $(\alpha^T C_P)_{j-\frac{1}{2}}$ are evaluated for simplicity at the beginning of the cycle. Note that C_P in Eqs. (2.17), (2.18) and (2.19) has units of $ergs/degree \cdot cm^3$ while c_P^i in Eq. (2.19) has units of $ergs/degree \cdot mole$. α^T is allowed to have a power-law dependence on c_{tot} and T in the same manner as D_i in Eq. (2.7).

Finally, the equation determining the change in Δr_j is

$$\Delta r'_j = \Delta r_j + \Delta t \left(v'_{j+\frac{1}{2}} - v'_{j-\frac{1}{2}} \right) \quad (2.22)$$

Differential Equations

The set of equations (2.1), (2.9), (2.13) and (2.22), together with the equations defining the various other terms in these equations, gives a set of implicit relations defining $c'_{i,j}$, T'_j , $\Delta r'_j$ and $v'_{j-\frac{1}{2}}$ in terms of $c_{i,j}$, T_j , Δr_j and $v_{j-\frac{1}{2}}$. We next show that these difference equations reduce to the appropriate differential equations in the limit $(\Delta r, \Delta t)$ approach zero.

From Eq. (2.22), we get

$$\frac{\Delta r}{\Delta r'} = 1 - \Delta t \frac{\partial v'}{\partial x} \quad (3.1)$$

as $\Delta r \rightarrow 0$. Now consider Eq. (2.1) in the limit that Δr , but not Δt , approaches 0.

Assume that quantities that are primed, i.e., refer to time $t + \Delta t$, are located at position x . Then the unprimed quantities are defined at $\underline{x} = x - v'\Delta t$, since after time Δt they will have moved from \underline{x} to x at velocity v' . To collect terms to order Δt , quantities defined at \underline{x} are expanded to first order in Δt . Thus for any quantity $Y(\underline{x})$ we have

$$Y(\underline{x}) = Y(x) - v'\Delta t \frac{\partial Y(x)}{\partial x} \quad (3.2)$$

Using Eq. (3.1), the left side of Eq. (2.1) becomes

$$c'_i - \left(1 - \Delta t \frac{\partial v'}{\partial x}\right) \left(c_i - v'\Delta t \frac{\partial c_i}{\partial x}\right)$$

or, to first order in Δt

$$c'_i - c_i + v'\Delta t \frac{\partial c_i}{\partial x} + \Delta t c_i \frac{\partial v'}{\partial x}.$$

We note the right side of Eq. (2.1) is already first order in Δt . Using the above expansion for the left side and dividing by Δt , we get

$$\frac{Dc_i}{Dt} = -c_i \frac{\partial v}{\partial x} + \sum_k R_{i,k} - \frac{\partial F_i^c}{\partial x}, \quad (3.3)$$

where

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + v \frac{\partial}{\partial x} \quad (3.4)$$

describes the time rate of change of a quantity in the frame moving with the fluid. Equation (2.4) has the limit

$$F_i^c = -\rho D_i \frac{\partial c_i}{\partial x} \frac{1}{\rho} \quad (3.5)$$

One can easily verify that Eq. (3.3) is consistent with forms given in standard texts such as Williams.[3]

For the acceleration equation, we let

$$m_{j-\frac{1}{2}} = \rho_{j-\frac{1}{2}} \Delta r_{j-\frac{1}{2}} \quad (3.6)$$

with

$$\Delta r_{j-\frac{1}{2}} = \frac{1}{2}(\Delta r_{j-1} + \Delta r_j) \quad (3.7)$$

define $\rho_{j-\frac{1}{2}}$. Then taking the limit $\Delta r \rightarrow 0$ and expanding to first order in Δt we get

$$v' - v + v' \Delta t \frac{\partial v}{\partial x} = - \frac{\Delta t}{\rho} \frac{\partial(P+Q)}{\partial x}.$$

As $\Delta t \rightarrow 0$ we have

$$\frac{Dv}{Dt} = -\frac{1}{\rho} \frac{\partial(P+Q)}{\partial x}, \quad (3.8)$$

with

$$Q = -\lambda \frac{\partial v}{\partial x}. \quad (3.9)$$

For the energy equation, the left side of Eq. (2.13) to first order in Δt is

$$\Delta r' \left[\mathcal{E}' - \left(\mathcal{E} - v' \Delta t \frac{\partial \mathcal{E}}{\partial x} \right) \left(1 - \Delta t \frac{\partial v'}{\partial x} \right) + (P+Q) \Delta t \frac{\partial v'}{\partial x} \right]$$

Inserting in Eq. (2.13), dividing by $\Delta r'$ and Δt gives

$$\frac{D\mathcal{E}}{Dt} = -[\mathcal{E} + (P+Q)] \frac{\partial v}{\partial x} - \frac{\partial F^\mathcal{E}}{\partial x}, \quad (3.10)$$

with

$$F^\mathcal{E} = \sum_i h_i F_i^c - \alpha^T C_P \frac{\partial T}{\partial x}. \quad (3.11)$$

Since it is sometimes interesting to consider T as the fundamental variable rather than \mathcal{E} , we also write out the differential equation for T . We note

$$\frac{D\mathcal{E}}{Dt} = \frac{D}{Dt} \sum_i c_i \epsilon_i = \sum_i \epsilon_i \frac{Dc_i}{Dt} + C_V \frac{DT}{Dt} \quad (3.12)$$

where

$$C_V = \sum_i c_i \frac{\partial \mathcal{E}_i}{\partial T} \quad (3.13)$$

Multiplying Eq. (3.3) by ϵ_i and summing over species we get

$$\sum_i \epsilon_i \frac{Dc_i}{Dt} = -\mathcal{E} \frac{\partial v}{\partial x} + \frac{1}{2} \sum_i \sum_k \epsilon_i R_{i,k} - \sum_i \epsilon_i \frac{\partial F_i^c}{\partial x} \quad (3.14)$$

The double sum can be written

$$\sum_i \sum_k \epsilon_i R_{i,k} = \sum_k Q_k \mathcal{R}_k, \quad (3.15)$$

where Q_k is the energy absorbed by the reaction k proceeding at rate \mathcal{R}_k . For example, the reaction described by Eq. (2.3) gives

$$\sum_i \epsilon_i R_{i,k} = -(\epsilon_a + \epsilon_b) \mathcal{R}_k.$$

Thus

$$Q_k = \epsilon_c + \epsilon_d - \epsilon_a - \epsilon_b$$

Using Eqs. (3.12), (3.14) and (3.15) in Eq. (3.10) gives

$$\frac{DT}{Dt} = \frac{1}{C_V} \left\{ - \sum_k Q_k \mathcal{R}_k - (P + Q) \frac{\partial v}{\partial x} - \frac{\partial F^{\mathcal{E}}}{\partial x} + \sum_i \epsilon_i \frac{\partial F_i^c}{\partial x} \right\} \quad (3.16)$$

If desired, the last two terms can be written to display their dependence on the gradient in temperature. The above form is the most convenient for our purposes.

Even though it is not used to actually update temperature, we use a finite difference analog of Eq. (3.16) to edit the sources of change in temperature during the running of problems. The finite difference form is derived similarly to Eq. (3.16) but using the finite difference Eqs. (2.1) and (2.13) instead of the differential Eqs. (3.3) and (3.16). There is a question of the time centering of ϵ_i when deriving the analog to Eq. (3.14). We multiply Eq. (2.1) by ϵ_i , not ϵ'_i . This results in the finite difference expression using Q_k calculated in terms of the ϵ_i at the beginning of the cycle.

Solution Technique

In discussing the method of solution for Eqs. (2.1), (2.9), (2.13), and (2.22), it is convenient to cast the problem into the more general vector form.

For each zone j in the problem, define a vector \vec{x}_j with $N+3$ components, where N is the number of species present. Let the first N components of \vec{x}_j be $c_{i,j}$, followed by T_j , Δr_j and $v_{j-\frac{1}{2}}$. The $N+3$ component is the only boundary-centered quantity, so placing it last in the vector can sometimes simplify the coding.

We define vectors \vec{f}_j by inserting Eq. (2.1) for each i in the first n components, and Eqs. (2.13), (2.22) and (2.9) in the next three components respectively. Transferring the right sides of these equations to the left, the set of equations becomes the vector equation

$$\vec{f}'_j = 0. \quad (4.1)$$

The prime on \vec{f}'_j means that it involves terms evaluated at the end of the cycle as well as the beginning.

We next observe that each \vec{f}'_j component involves \vec{x}'_j from its own zone j or one of its neighbors, thus Eq. (4.1) is

$$\vec{f}_j(\vec{x}'_{j-1}, \vec{x}'_j, \vec{x}'_{j+1}) = 0.$$

Equation (4.1) is solved by a modified Newton's iteration scheme. One makes a guess \vec{x}^n_j for \vec{x}'_j , expands Eq. (4.1) to first order in the correction, and solves the resulting set of linear equations for the correction. We actually solve for the relative correction

$$\delta \vec{y}_j = \frac{\delta \vec{x}_j}{\vec{x}_j} \quad (4.2)$$

which has the advantage of being a dimensionless number. Thus let

$$\vec{y}_j = \ln \vec{x}_j, \quad (4.3)$$

so that

$$\delta \vec{x}_j \frac{\partial \vec{f}_i}{\partial \vec{x}_j} = \frac{\partial \vec{x}_j}{\vec{x}_j} \left(\vec{x}_j \frac{\partial \vec{f}_i}{\partial \vec{x}_j} \right) = \delta \vec{y}_j \frac{\partial \vec{f}_i}{\partial \vec{y}_j}$$

$\frac{\partial \vec{f}_i}{\partial \vec{y}_j}$ is a quantity with the same dimensions as \vec{f}_j . This makes the coding much simpler, since the derivatives are now a dimensionless multiple of the functions themselves. Then Eq. (4.1) becomes

$$\vec{f}_j^n + \delta \vec{y}_{j-1}^n \frac{\partial \vec{f}_j^n}{\partial \vec{y}_{j-1}} + \delta \vec{y}_j^n \frac{\partial \vec{f}_j^n}{\partial \vec{y}_j} + \delta \vec{y}_{j+1}^n \frac{\partial \vec{f}_j^n}{\partial \vec{y}_{j+1}} = 0. \quad (4.4)$$

The superscript n means that primed quantities are evaluated using the n^{th} guess at their final values.

One notices that Eq. (4.4) is a tridiagonal matrix equation for a vector whose components are themselves vectors. Thus let \vec{X} be the vector whose first $N+3$ components are \vec{x}_1 , the next $N+3$ are \vec{x}_2 , and so forth. A similar construction gives $\delta \vec{Y}$ and \vec{F} . In matrix form Eq. (4.1) is

$$\vec{F}' = 0$$

and Eq. (4.4) becomes

$$T^n \delta \vec{Y}^n = \vec{F}^n, \quad (4.5)$$

where

$$T = \begin{vmatrix} A_1 & B_1 & & & \\ C_2 & A_2 & B_2 & & \\ & C_3 & A_3 & B_3 & \\ & & & \ddots & \ddots & \ddots \\ & & & & \ddots & \ddots & \ddots \end{vmatrix},$$

$$\delta \vec{Y} = \begin{vmatrix} \delta \vec{y}_1 \\ \delta \vec{y}_2 \\ \delta \vec{y}_3 \\ \vdots \\ \vdots \end{vmatrix},$$

and

$$\vec{F} = \begin{vmatrix} \vec{f}_1 \\ \vec{f}_2 \\ \vec{f}_3 \\ \vdots \\ \vdots \end{vmatrix}.$$

To improve the numerical behavior of the routines used to solve the matrix Eq. (4.5), we normalize the individual equations so that the largest term in any equation is one. This is done by summing the absolute values of the contributions to the right side of Eq. (4.5) whenever the matrix T^n is calculated. We then divide each row of the matrix by the corresponding sum.

We use Gaussian elimination in the form of a standard LU decomposition of T to solve Eq. (4.5). That is, we find matrices L and U such that L is block-lower triangular and U is unit-upper triangular and

$$T = LU, \tag{4.6}$$

$$L = \begin{vmatrix} D_1 & & & & \\ C_2 & D_2 & & & \\ & C_3 & D_3 & & \\ & & & \ddots & \\ & & & & \ddots & \ddots \end{vmatrix},$$

$$U = \begin{vmatrix} I & E_1 & & & \\ & I & E_2 & & \\ & & I & E_3 & \\ & & & \ddots & \\ & & & & \ddots & \end{vmatrix}.$$

For any \vec{x} , $L\vec{x} = \vec{y}$ or $U\vec{x} = \vec{y}$ is easily solved. Then for Eq. (4.5)

$$LU\delta\vec{Y} = \vec{F}$$

is solved by back substitution in two steps. First

$$L\vec{W} = \vec{F}$$

is solved for \vec{W} , then solving

$$U\delta\vec{Y} = \vec{W}$$

gives $\delta\vec{Y}$.

The algorithm[4] for performing the LU decomposition is

$$\begin{aligned} D_1 &= A_1 \\ E_k &= D_k^{-1}B_k \\ D_k &= A_k - C_k E_{k-1} \text{ when } k > 1. \end{aligned} \tag{4.7}$$

Note that the subdiagonal elements of L are the same as the subdiagonal elements of T . Of course, Eqs. (4.7) are matrix equations themselves. When the inverse is needed in one of these quantities, an LU decomposition is done on these matrices. The routine used in this case is a standard library routine.[5] The explicit inverse of a matrix is never actually calculated. Rather, its LU decomposition is, and when the inverse is present in an expression we want to evaluate, the particular set of linear equations it represents is solved.

It is interesting to point out how much work is involved in performing the LU decomposition of T . The block matrices in T are of dimension $M = N + 3$. The order of the number of operations to do one LU decomposition of an M -by- M matrix is M^3 . This is also the order of the number of operations to multiply two matrices together. Thus the evaluation of D_k in Eqs. (4.7) is order M^3 . To evaluate E_k involves two order- M^3 operations, a LU decomposition of D_k and the solution of the M vector equations needed to get the M columns of E_k . Thus we are doing a total of three order- M^3 operations for each zone. If our original matrix T were block diagonal, we would be doing one order- M^3 operation per zone to solve the system. Thus to couple the nearest neighbors in our difference scheme has made roughly a factor-of-three difference in the time needed to solve one cycle.

Our system would have been block diagonal if we had operator-split the chemistry from the hydrodynamics. In general, we expect to pick up more than a factor of three in timestep by going from an operator-split scheme to a fully coupled scheme. This has been confirmed in actual calculations.

Once we have calculated the LU decomposition of T^n in Eq. (4.5), calculating the components of $\delta\vec{Y}^n$ becomes a matter of adding different multiples of the components of \vec{F}^n together. One can look on Eq. (4.5) as a prescription for $\delta\vec{Y}^n$ in terms of a weighted sum of elements of \vec{F}^n . In particular, if one used other matrices T' that didn't differ much from T^n , one would expect to have an iterative scheme that also would be satisfactory. In solving Eq. (4.5), the LU decomposition is the most expensive operation, being order M more expensive than solving the system by back substitution. To save computer time we will first try to solve Eq. (4.4) using the last available T^m . That is, we solve

$$T^m \delta\vec{Y}^n = \vec{F}^n \quad (4.8)$$

to get a new $\delta\vec{Y}^n$. If this $\delta\vec{Y}^n$ appears to be converging to zero, we continue to use the last available T^m without doing more LU decompositions of the full system.

We assume \vec{Y} has converged if

$$\max\{\delta Y_i\} < \epsilon, \quad (4.9)$$

where ϵ is typically 10^{-6} . It is not clear how small ϵ must be for stability or accuracy, and it may be possible to use considerably larger ϵ in many problems.

In the case of velocity, requiring

$$\frac{\delta v_{j-\frac{1}{2}}^n}{v_{j-\frac{1}{2}}^n} < \epsilon$$

may be unreasonable. As pointed out previously, if the system is in pressure equilibrium, $v_{j-\frac{1}{2}}\Delta t$ is the distance the boundary between zones $j-1$ and j has moved to maintain pressure equilibrium. If the resolution of Δr_{j-1} and Δr_j is ϵ , then it is more reasonable just to assure that

$$\delta v_{j-\frac{1}{2}}\Delta t < \epsilon \cdot \min\{\Delta r_{j-1}, \Delta r_j\}. \quad (4.10)$$

In fact, additional resolution in $\delta v_{j-\frac{1}{2}}$ can be lost in machine roundoff, since the code calculates $\delta v_{j-\frac{1}{2}}$ as a difference between Δr_{j-1} and Δr_j . We use Eq. (4.10) to determine convergence in velocity.

There is another anomaly associated with velocity. It is the only variable in \vec{X} that can physically be zero. Using zero in Eq. (4.2) causes the computer to try to divide by zero. Thus when any guess $v_{j-\frac{1}{2}}^n$ is zero, we make $v_{j-\frac{1}{2}}^n$ finite by using

$$v_{j-\frac{1}{2}}^n\Delta t = \epsilon_r\Delta r_j^n, \quad (4.11)$$

where $\epsilon_r \ll \epsilon$ is on the order of machine roundoff. if $v'_{j-\frac{1}{2}}$ is truly zero, this will result in a $\delta v_{j-\frac{1}{2}}^n$ that satisfies Eq. (4.10), and thus does not affect the convergence test.

It will sometimes happen that the solution of Eq. (4.5), the linearized version of Eq. (4.1), will predict a $x^n = c_{i,j}$, T_j or Δr_j that is zero or negative. We know from physical considerations that these quantities should never be negative or zero. If

$$x^{n+1} = x^n \left(1 + \frac{\delta x^n}{x^n} \right) \quad (4.12)$$

predicts $x^{n+1} \leq 0$, we use

$$x^{n+1} = \frac{x^n}{1 - \frac{\delta x^n}{x^n}} \quad (4.13)$$

Equations (4.12) and (4.13) are identical to first order in $\delta x^n/x^n$. This procedure guarantees the use of physically acceptable x^n . If this is not done it is possible to find mathematically correct but nonphysical solutions to difference equations. This occurs because these equations are nonlinear.

The initial guess at x' is made from a second-order fit in Δt to the past history. The past history is saved in the form of Newton's interpolating polynomials so that

$$x(t) = [x_0] + (t - t_0)[x_0, x_{-1}] + (t - t_0)(t - t_{-1})[x_0, x_{-1}, x_{-2}] \quad (4.14)$$

where

$$\begin{aligned} [x_0] &= x(t_0) \\ [x_0, x_{-1}] &= \frac{[x_0] - [x_{-1}]}{t_0 - t_{-1}} \\ [x_0, x_{-1}, x_{-2}] &= \frac{[x_0, x_{-1}] - [x_{-1}, x_{-2}]}{t_0 - t_{-2}} \end{aligned}$$

In all variables except velocity, a check is made to see if the second-order polynomial gives a negative interpolation. If it does, a first-order interpolation is tried. If that is also negative, an interpolation analogous to Eq. (4.13) is used.

To determine whether a new LU decomposition of T is needed, a check is made on the rate of convergence by calculating a norm

$$\|\delta\vec{Y}^n\| = \sum_m (\delta Y_m^n)^2, \quad (4.15)$$

where the index m runs over all δY_m that have not converged. For m corresponding to velocity components, Eq. (4.10) suggests we use

$$\delta Y_m = \frac{\delta v_{j-\frac{1}{2}} \Delta t}{\min\{\Delta r_{j-1}, \Delta r_j\}} \quad (4.16)$$

If an iteration using an old T^m in Eq. (4.8) satisfies

$$\|\delta\vec{Y}^n\| > \|\delta\vec{Y}^{n-1}\|,$$

we suspect that the old T^m has moved us away from, rather than toward, the true solution. In that case, we discard $\delta\vec{Y}^n$, let $\vec{Y}^{n+1} = \vec{Y}^n$, and update the matrix T^m for the next iteration.

If

$$\|\delta\vec{Y}^n\| > f\|\delta\vec{Y}^{n-1}\|, \quad (4.17)$$

where f is some input fraction of 1, we update \vec{Y}^{n+1} using $\delta\vec{Y}^n$ but require a new LU decomposition of T for the next iteration. We argue that although the iterations are converging, we expect a much faster rate of convergence with a new T . f is typically a number like 1/4.

Finally, we allow an old LU decomposition of T to be used no more than a fixed number of times. It occasionally happens that the iterations will

converge, but on an incorrect solution to the true equations. This maximum number of uses of an old T^m is usually set at problem generation to some number like 20.

At generation we also specify the maximum number of iterations allowed to solve Eq. (4.1). If the iterations do not converge in this number of iterations, we decrease the timestep Δt by a factor of 2 and start the whole cycle over. This can happen when there is a sudden change in the problem, as in the onset of rapid burn. In that case the linearized Eq. (4.5) differs considerably from the nonlinear Eq. (4.1). The maximum number of iterations allowed can vary with the problem and with the frequency at which the LU decomposition is updated. For a complicated problem that is averaging 5 or so iterations per cycle, a maximum of 20 iterations per cycle is not unreasonable, especially if one is using old LU decompositions most of the time.

Eulerian Calculations

HCT can do a remap of variables defined on one grid so that they are defined on a second. An Eulerian calculation is done by remapping back onto a fixed grid after each cycle. Thus the grid remains the same throughout the calculation. The grid can also be determined dynamically throughout the problem by a method that will be described later. Since this remapping determines how much material moves from one zone to another during a cycle, it corresponds to the advection terms of the Eulerian formulation.

The remap is limited in the sense that each zone of the original grid must overlap the corresponding zone in the new grid. This means that the new boundary at $j - \frac{1}{2}$ between zones $j - 1$ and j must be in either the original zone $j-1$ or zone j . Let Δr_j be the zone width of the j^{th} zone in the original grid. Then let $\Delta r'_j$ be the zone width of the j^{th} zone in the new grid. Then

$$\Delta r'_j = \Delta r_j + \Delta x_{j+\frac{1}{2}} - \Delta x_{j-\frac{1}{2}}, \quad (5.1)$$

where Δx is the distance a boundary is to be moved.

There are two types of quantities that are to be remapped: zone-centered and boundary-centered quantities.

For any zone-centered quantity y_j , Eq. (5.1) implies

$$\Delta r'_j y'_j = \Delta r_j y_j + \Delta x_{j+\frac{1}{2}} y_{j+\frac{1}{2}} - \Delta x_{j-\frac{1}{2}} y_{j-\frac{1}{2}} \quad (5.2)$$

We are left, then, with the choice for $y_{j-\frac{1}{2}}$. This choice determines the formal order of accuracy of the remap.

Note that Eq. (5.2) will conserve the integral of y across the grid (except possibly at boundaries). Since energy should be conserved by the remap rather than temperature, temperature is determined implicitly after the remap of energy.

The choice of lowest order accuracy is to take $y_{j-\frac{1}{2}}$ equal to the value of y_j in the zone $\Delta x_{j-\frac{1}{2}}$ overlaps. This gives first-order accuracy, and in many situations gives rise to large amounts of numerical diffusion. This choice is sometimes called donor cell or upwind differencing, since the quantity being advected into a zone has its value determined by the zone from which it comes. This choice can be written

$$y_{j-\frac{1}{2}} = \left[1 - \alpha_{j-\frac{1}{2}}\right] y_{j-1} + \alpha_{j-\frac{1}{2}} y_j, \quad (5.3)$$

where

$$\alpha_{j-\frac{1}{2}} = \frac{1}{2} \left(1 + \frac{\Delta x_{j-\frac{1}{2}}}{|\Delta x_{j-\frac{1}{2}}|} \right) \quad (5.4)$$

is either 0 or 1.

Second-order accuracy can be obtained by interpolating $y_{j-\frac{1}{2}}$ between y_{j-1} and y_j . Thus

$$y_{j-\frac{1}{2}} = \left[1 - \beta_{j-\frac{1}{2}}\right] y_{j-1} + \beta_{j-\frac{1}{2}} y_j, \quad (5.5)$$

with

$$\beta_{j-\frac{1}{2}} = \frac{2}{\Delta r_{j-1} + \Delta r_j} \left\{ \frac{1}{2} \Delta r_{j-1} + \eta \Delta x_{j-\frac{1}{2}} \right\} \quad (5.6)$$

The term proportional to η gives a degree of “upwindness” to the weighting. We use $\eta = 1$, since for this choice $\beta_{j-\frac{1}{2}} = 1$ if $\Delta x_{j-\frac{1}{2}} = \Delta r_j/2$.

With second-order remaps it is possible to advect more out of a zone than was present in the beginning, giving rise to negative densities. This occurs because the remap takes $y_{j-\frac{1}{2}} \Delta x_{j-\frac{1}{2}}$ out of a zone containing $y_j \Delta r_j$, when it is possible that $y_{j-\frac{1}{2}} \gg y_j$. We perform a check to see if this is going to happen, and if it is we do a first-order remap of this quantity.

It turns out that in combustion problems some of the $c_{i,j}$ have very large gradients, often changing an order of magnitude or more from zone to zone. In such a circumstance, neither the first-order remap nor the second-order one described above is adequate. This shows up in the running of a problem when the Lagrange cycle changes are nearly cancelled by the following remap. The problem is that these remaps associate too large a quantity with the boundary. For this case we have another choice that averages inverses of y_j to get $y_{j-\frac{1}{2}}$. Thus

$$\frac{1}{y_{j-\frac{1}{2}}} = \frac{1 - \beta_{j-\frac{1}{2}}}{y_{j-1}} + \frac{\beta_{j-\frac{1}{2}}}{y_j} \quad (5.7)$$

This expression has second-order accuracy when y_{j-1} is nearly equal to y_j , since Eq. (5.7) gives

$$y_{j-\frac{1}{2}} = \left[1 - \beta_{j-\frac{1}{2}}\right] y_{j-1} + \beta_{j-\frac{1}{2}} y_j$$

plus terms of order $(y_j - y_{j-1})$. But it has the advantage that it takes $y_{j-\frac{1}{2}}$ to be the minimum of y_{j-1}, y_j rather than the average when there are large gradients.

For the first-order remap, the total zone energy is remapped in the same manner as the species concentrations. For the second-order remap defined by Eq. (5.5), we map over the energy carried by each species separately. That is, we write

$$(c_i \epsilon_i)_{j-\frac{1}{2}} = \left[1 - \beta_{j-\frac{1}{2}}\right] (c_i \epsilon_i)_{j-1} + \beta_{j-\frac{1}{2}} (c_i \epsilon_i)_j \quad (5.8)$$

This is done so that if we drop to first order in any one species, the corresponding fraction of the zone energy will be mapped over correctly.

For the remap defined by Eq. (5.7) we need an average energy associated with $(c_i)_{j-\frac{1}{2}}$. We define

$$(\epsilon_i)_{j-\frac{1}{2}} = \left[1 - \beta_{j-\frac{1}{2}}\right] (\epsilon_i)_{j-1} + \beta_{j-\frac{1}{2}} (\epsilon_i)_j \quad (5.9)$$

and then use

$$(c_i \epsilon_i)_{j-\frac{1}{2}} = (c_i)_{j-\frac{1}{2}} (\epsilon_i)_{j-\frac{1}{2}}. \quad (5.10)$$

This gives a minimum c_i but associates it with the temperature at the boundary.

With the total energy and species concentrations of the zone determined from Eq. (5.2), we next determine the new temperature that gives that energy. This gives the new temperature to be associated with the new grid.

Velocity is the only boundary-centered quantity to be determined. Since momentum is a conserved quantity associated with velocity, we remap momentum, where

$$p_{j-\frac{1}{2}} = m_{j-\frac{1}{2}} v_{j-\frac{1}{2}} \quad (5.11)$$

and $m_{j-\frac{1}{2}}$ is given by Eq. (2.10). The analogue of Eq. (5.2) is

$$\Delta r'_{j-\frac{1}{2}} p'_{j-\frac{1}{2}} = \Delta r_{j-\frac{1}{2}} p_{j-\frac{1}{2}} + \Delta x_j p_j - \Delta x_{j-1} p_{j-1} \quad (5.12)$$

where the only difference is in the centering of these terms. We take

$$\Delta r_{j-\frac{1}{2}} = \frac{1}{2} \{ \Delta r_{j-1} + \Delta r_j \} \quad (5.13)$$

$$\Delta x_j = \frac{1}{2} \{ \Delta x_{j-\frac{1}{2}} + \Delta x_{j+\frac{1}{2}} \} \quad (5.14)$$

For the first-order remap we have

$$p_j = [1 - \alpha_j] p_{j-\frac{1}{2}} + \alpha_j p_{j+\frac{1}{2}} \quad (5.15)$$

with

$$\alpha_j = \frac{1}{2} \left(1 + \frac{\Delta x_j}{|\Delta x_j|} \right). \quad (5.16)$$

Δx_j is given in terms of the fundamental $\Delta x_{j-\frac{1}{2}}$ by Eq. (5.14).

For the second-order remap we write

$$p_j = [1 - \beta_j] p_{j-\frac{1}{2}} + \beta_j p_{j+\frac{1}{2}} \quad (5.17)$$

with

$$\beta_j = \frac{1}{\Delta r_j} \left\{ \frac{1}{2} \Delta r_{j-\frac{1}{2}} + \eta \Delta x_j \right\}, \quad (5.18)$$

where $\Delta r_{j-\frac{1}{2}}$ and Δx_j are given by Eq. (5.13) and Eq. (5.14).

With momentum we do not allow the second-order remap for $p_{j-\frac{1}{2}}$ implied by Eq. (5.7), since there are not usually such large gradients in velocity present. Equation (5.7) is only needed for quantities that vary by an order of magnitude or so from zone to zone, and only then if the physics tends to negate the remap changes in the next Lagrange cycle.

Dynamic Rezoning

HCT has a option for determining $\Delta r'_j$ dynamically in such a way as to concentrate zoning in areas of high temperature gradients. High temperature gradients are associated with high rates of energy release. High rates of energy release are also associated with rapid changes in species concentrations. Thus this option will usually result in better resolution of the entire problem.

The grid spacing is determined by the choice of $\Delta x_{j-\frac{1}{2}}$ in Eq. (5.1). In a normal Eulerian calculation with a fixed grid one chooses

$$\Delta x_{j-\frac{1}{2}} = -v'_{j-\frac{1}{2}} \Delta t. \quad (6.1)$$

This just moves the grid back to the spacing that existed before the Lagrange calculation.

To place zoning where the temperature gradient is large, we could choose Δx_j so that

$$\left(\Delta r \frac{\partial T}{\partial x} \right)_j = |\Delta T_j| = \text{constant}.$$

However, this would result in unusually large zones where $\partial T / \partial x$ is small. In general, we want to satisfy this condition where gradients are large, but have the zoning grow slowly as we move away from high-gradient regions. To do this we impose the additional constraint of minimizing zone-size mismatches across the grid.

We formulate the problem as follows: Let $y(x)$ be a continuous function whose value at x_j is the desired zone thickness there. Let $f(x)$ be proportional to the absolute value of $\partial T / \partial x$, and let

$$z = yf = \alpha y \frac{\partial T}{\partial x} \quad (6.2)$$

where α is a constant. Then the problem is to determine y such that

$$I(y) = \int_L \frac{dx}{y} \left\{ (y')^2 + z^2 \right\} \quad (6.3)$$

$$y' = \frac{dy}{dx}$$

is a minimum subject to the constraint

$$N = \int_L \frac{dx}{y} \quad (6.4)$$

Equation (6.4) just requires that there be N zones in the grid. Since dy/dx is the zone mismatch between adjacent zones, minimizing the contribution of the first term in Eq. (6.3) minimizes zoning mismatches across the whole grid. Minimizing the contribution of z^2 to $I(y)$ will keep y small in regions of large temperature gradients. The proportionality constant α in Eq. (6.2) is chosen so that

$$y \left| \frac{dT}{dx} \right|_{max} = \frac{1}{N^*} \int_L dx \left| \frac{dT}{dx} \right| \quad (6.5)$$

Thus N^* is the minimum number of zones reserved to describe the temperature change across the grid.

The problem of minimizing $I(y)$ subject to Eq. (6.4) is solved using the Euler-Lagrange equations. We make the change of variables

$$y = w^2 \quad (6.6)$$

so that the resulting differential equation has no first-derivative terms. The Euler-Lagrange prescription gives[6]

$$w'' + \frac{1}{4} \left\{ \lambda - z^2 \right\} \frac{1}{w^3} = 0, \quad (6.7)$$

with the boundary conditions

$$w'(0) = w'(L) = 0 \quad (6.8)$$

λ is chosen to satisfy Eq. (6.4).

Equations (6.7) and (6.4) represent a nonlinear eigenvalue problem. If Eq. (6.7) were linear, its solution would present no difficulty. This suggests we assume a reasonable first guess for w_0 , λ_0 , linearize Eq. (6.7) about this guess, and solve for the correction. Let Eq. (6.7) be written as

$$F(w'', w, \lambda) = 0. \quad (6.9)$$

Linearizing gives

$$\mathcal{L}\delta w = -F_0 - \frac{\partial F}{\partial \lambda}\delta\lambda, \quad (6.10)$$

$$\delta w'(0) = \delta w'(L) = 0,$$

with

$$L \equiv \frac{\partial F}{\partial w''} \frac{\partial^2}{\partial x^2} + \frac{\partial F}{\partial w} \quad (6.11)$$

\mathcal{L} is a linear operator. We have assumed w_0 satisfies the boundary conditions of Eq. (6.8).

The solution

$$\delta w = -\mathcal{L}^{-1} \left\{ F_0 + \frac{\partial F}{\partial \lambda} \delta\lambda \right\} \quad (6.12)$$

depends on $\delta\lambda$. This allows us to choose $\delta\lambda$ so that the constraint, Eq. (6.4), is satisfied, i.e.

$$\int \frac{dx}{w_0^3} \delta w = - \int \frac{dx}{w_0^3} \mathcal{L}^{-1} \left\{ F_0 + \frac{\partial F}{\partial \lambda} \delta\lambda \right\} = 0$$

implies

$$\delta\lambda = \frac{- \int \frac{dx}{w_0^3} \mathcal{L}^{-1} F_0}{\int \frac{dx}{w_0^3} \mathcal{L}^{-1} \frac{\partial F}{\partial \lambda}} \quad (6.13)$$

We have assumed that w_0 already satisfies Eq. (6.4).

This method should work well when δw is small compared to w . To allow for the possibility that δw is large, we want a method that insures δw is small, and allows us to arrive at the solution to Eq. (6.7) by a series of small corrections to the latest values of w . Let us parameterize the small changes in w by the change in another parameter ξ . Let

$$\delta w(x, \xi) = \frac{\partial w(x, \xi)}{\partial \xi} \Delta \xi, \quad (6.14)$$

where, by definition, $\Delta \xi = 1$ gives the δw of Eq. (6.10). From Eq. (6.12) δw is proportional to F_0 and $\partial F / \partial \lambda \delta \lambda$. Replace F_0 by $F_0 \Delta \xi$ and let

$$\delta \lambda(\xi) = \frac{\partial \lambda(\xi)}{\partial \xi} \Delta \xi. \quad (6.15)$$

Then the right side of Eq. (6.12) can be made as small as desired. Replacing Eq. (6.10) we have

$$\mathcal{L} \frac{\partial w}{\partial \xi} = -F_0 - \frac{\partial F}{\partial \lambda} \frac{\partial \lambda}{\partial \xi} \quad (6.16)$$

$$\frac{\partial w'(0, \xi)}{\partial \xi} = \frac{\partial w'(L, \xi)}{\partial \xi} = 0,$$

which can be used to update w for arbitrarily small $\Delta \xi$ using

$$w(x, \xi) = w_0(x) + \int_0^\xi d\xi' \frac{\partial w(x, \xi')}{\partial \xi'} \quad (6.17)$$

$\partial \lambda / \partial \xi$ is evaluated by the generalization of Eq. (6.13):

$$\frac{\partial \lambda}{\partial \xi} = \frac{-\int \frac{dx}{w^3} \mathcal{L}^{-1} F_0}{\int \frac{dx}{w^3} \mathcal{L}^{-1} \frac{\delta F}{\delta \lambda}} \quad (6.18)$$

Then

$$\lambda(\xi) = \lambda_0 + \int_0^\xi d\xi' \frac{\partial \lambda(\xi')}{\partial \xi'} \quad (6.19)$$

gives λ for any ξ .

Note that Eqs. (6.16) and (6.11) give

$$\frac{\partial F}{\partial w''} \frac{\partial w''}{\partial \xi} + \frac{\partial F}{\partial w} \frac{\partial w}{\partial \xi} + \frac{\partial F}{\partial \lambda} \frac{\partial \lambda}{\partial \xi} \equiv \frac{dF}{d\xi} = -F_0. \quad (6.20)$$

Integrating over ξ give

$$\int_0^\xi d\xi' \frac{dF(\xi')}{d\xi'} = F(\xi) - F_0 = -\xi F_0.$$

This solves Eq. (6.9) when $\xi = 1$. This shows that Eqs. (6.16), (6.17), (6.18) and (6.19) provide a generalization of the one-step linearization procedure represented by Eqs. (6.10) and (6.13).

In discrete form we write Eq. (6.16) as

$$\mathcal{L}^n \frac{\partial w^{n+\frac{1}{2}}}{\partial \xi} = -F_0 - \frac{\partial F^n}{\partial \lambda} \frac{\partial \lambda^{n+\frac{1}{2}}}{\partial \xi} \quad (6.21)$$

where n indicates the successive values of ξ , i.e.

$$\xi^{n+1} = \xi^n + \Delta \xi^{n+\frac{1}{2}}, \quad (6.22)$$

and Eqs. (6.17) and (6.19) give the prescription for updating w_j and λ .

$$w_j^{n+1} = w_j^n + \frac{\partial w_j^{n+\frac{1}{2}}}{\partial \xi} \Delta \xi^{n+\frac{1}{2}}, \quad (6.23)$$

$$\lambda^{n+1} = \lambda^n + \frac{\partial \lambda^{n+\frac{1}{2}}}{\partial \xi} \Delta \xi^{n+\frac{1}{2}} \quad (6.24)$$

The index j on w_j^n denotes the spatial zoning. Thus in Eq. (6.21) \mathcal{L} , F_0 and $\partial F^n / \partial \lambda$ are evaluated using the last available w_j and λ , and Eqs. (6.23) and (6.24) are used to update w_j and λ for the next cycle.

We use the obvious discrete form of the second derivative of w in \mathcal{L} , i.e.

$$\frac{\partial^2}{\partial x^2} \frac{\partial w_j}{\partial \xi} = \frac{1}{\Delta r_j} \left\{ \frac{\partial}{\partial x} \frac{\partial w_{j+\frac{1}{2}}}{\partial \xi} - \frac{\partial}{\partial x} \frac{\partial w_{j-\frac{1}{2}}}{\partial \xi} \right\} \quad (6.25)$$

Except at the problem boundaries, where the boundary conditions of Eq. (6.16) apply, one has

$$\frac{\partial}{\partial x} \frac{\partial w_{j-\frac{1}{2}}}{\partial \xi} = \frac{1}{\Delta r_{j-\frac{1}{2}}} \left\{ \frac{\partial w_j}{\partial \xi} - \frac{\partial w_{j-1}}{\partial \xi} \right\} \quad (6.26)$$

where $\Delta r_{j-\frac{1}{2}}$ is given by Eq. (3.7). The second-derivative operator in \mathcal{L} is the only part connecting adjacent spatial zones. Thus in discrete form \mathcal{L}

is a tridiagonal matrix and \mathcal{L}^{-1} is evaluated using the LU decomposition algorithm of Eqs. (4.7).

To evaluate $f(x)$ in Eq. (6.2), the temperature distribution is tabulated and the derivative of the table used. We first define

$$\frac{d\mathcal{T}}{dx} = \left| \frac{dT}{dx} \right| \quad (6.27)$$

through

$$\mathcal{T}_j = \mathcal{T}_{j-1} + |T_j - T_{j-1}| \quad (6.28)$$

and then define \mathcal{T} at the boundaries by

$$\mathcal{T}_{j-\frac{1}{2}} = \frac{1}{2} \{ \mathcal{T}_{j-1} + \mathcal{T}_j \}. \quad (6.29)$$

The numerical procedure for updating the grid then consists of the following steps:

First the temperature distribution is used to define \mathcal{T} on the existing zone boundaries. The initial guess at w is taken as

$$(w_0)_j = (\Delta r_j)^{\frac{1}{2}}.$$

One uses an initial value for λ and α from the last cycle. The operator \mathcal{L} can then be evaluated and Eq. (6.21) solved for $\partial w / \partial \xi$. One calculates a $\Delta \xi$ that changes $w(x)$ by no more than a fixed number, typically 0.05. Then w , λ are updated using Eqs. (6.23) and (6.24).

At this point the mesh is redefined in such a way that

$$\int_{\Delta r'_j} \frac{dx}{w^2} = \frac{1}{N} \int_L \frac{dx}{w^2} \quad (6.30)$$

$\Delta r'_j$ is the new mesh spacing. This is done with the conservative first-order remap of Eqs. (5.3) and (5.4), $1/w^2$ being the conserved quantity. This is done so that if w is changing rapidly as a function of ξ , the grid will always define w accurately. A conservative remap is done so that the new mesh will represent the same number of zones as the old.

This cycle is repeated until $\xi = 1$. Because of errors introduced by the discrete form of the original differential equations, errors have been introduced in w . We iterate the whole procedure again until the current iteration

agrees with the last iteration to within a convergence factor, currently 0.01. Between each iteration w is normalized so that Eq. (6.4) is satisfied.

Once one has w and λ for a fixed α , one can correct α so that Eq. (6.5) is satisfied. In analogy to Eq. (6.12) one gets

$$\delta w = -\mathcal{L}^{-1} \left\{ \frac{\partial F}{\partial \alpha} + \frac{\partial F}{\partial \lambda} \frac{\partial \lambda}{\partial \alpha} \right\} \delta \alpha \quad (6.31)$$

where $\partial \lambda / \partial \alpha$ is chosen to satisfy Eq. (6.4).

To do this, we need the location where $|dT/dx|$ is a maximum. We find the zone which has the largest $|dT/dx|$ and fit $|dT/dx|$ with a second-order polynomial through the value in this zone and its neighbors. We assume the zone values define $|dT/dx|$ at the centers of the zones. Thus if $f = |dT/dx|$ we have

$$f = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} f_1 + \dots, \quad (6.32)$$

or

$$f = (x - x_2)(x - x_3) \alpha_1 f_1 + \dots \quad (6.33)$$

Actually, if jx is the zone with the largest value of $|dT/dx|$, we let

$$\begin{aligned} x_1 &= -(\Delta r_{jx-1} + \Delta r_{jx}) \\ x_2 &= 0 \\ x_3 &= \Delta r_{jx} + \Delta r_{jx+1} \end{aligned} \quad (6.34)$$

This implies spacing that is twice the actual grid spacing, but the factors of 2 drop out in the quantities we want.

The x for which the polynomial is maximum is given by

$$\underline{x} = \frac{(x_2 + x_3) \alpha_1 f_1 + (x_1 + x_3) \alpha_2 f_2 + (x_1 + x_2) \alpha_3 f_3}{2 (\alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3)} \quad (6.35)$$

Since the same coefficients of Eq. (6.33) define the polynomial fit to any zone-centered grid quantity, we can calculate the effect a change δw in w will make in

$$y \left| \frac{dT}{dx} \right|_{max} = (\underline{x} - x_2) (\underline{x} - x_3) \alpha_1 \left(w_1^2 + 2w_1 \delta w_1 \right) \left| \frac{dT}{dx} \right|_1 + \dots \quad (6.36)$$

The terms proportional to δw are the correction terms. From Eq. (6.31) δw is proportional to $\delta\alpha$. $\delta\alpha$ is chosen so that δw will change Eq. (6.36) to satisfy Eq. (6.5).

One could iterate the procedure to get the exact value of α desired at that time, but in practice we calculate what α should be and only add a fraction (typically 0.1) of $\delta\alpha$ to the current value. Thus after around 10 cycles α approaches the desired value. We do not completely correct α because the polynomial fit to $|dT/dx|$ does not correspond to an exact description of the temperature, and thus there is some variation in the calculated α as the high-gradient region moves through a zone.

We also use this method to initialize the grid at the beginning of the problem to match an initial nonuniform variable distribution. In that case $f(x)$ may be known exactly. It is not necessary to have the original estimate of the zoning accurately known for the iteration procedure to converge if the average value of f over the zone is used whenever the zone centered value of f is needed. Thus

$$f_j = \frac{1}{\Delta r_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} dx f(x). \quad (6.37)$$

For example, if $f = |dT/dx|$ one uses

$$f_j = \frac{1}{\Delta r_j} \left| T(x_{j+\frac{1}{2}}) - T(x_{j-\frac{1}{2}}) \right|$$

rather than $f_j = |dT(x_j)/dx|$. This will allow f to reflect discontinuities in T even if the grid is not initially chosen to reflect them.

Kinetics Calculations

HCT reads reaction data from a disk file CDAT at the initiation phase of each problem. The constants associated with each reaction are stored in this file. As currently implemented, HCT uses the collection of data maintained by Westbrook and Chase,[7] which also includes the fits to the thermodynamic data.

The reactions in CDAT are classified into standard forms or types, which can be one of the following:

$a + b$	\leftrightarrow	$c + d,$	<i>I</i>
$a + b$	\leftrightarrow	$c + c,$	<i>II</i>
$a + M$	\leftrightarrow	$M + b + c,$	<i>III</i>
$a + M$	\leftrightarrow	$M + b + b,$	<i>IV</i>
a	\leftrightarrow	$b + c,$	<i>V</i>
a	\leftrightarrow	$b + b,$	<i>VI</i>
a	\leftrightarrow	$b,$	<i>VII</i>
$a + M$	\leftrightarrow	$b + M,$	<i>VIII</i>
a	\leftrightarrow	$b + c + d,$	<i>IX</i>
$a + b$	\rightarrow	$b + d + d,$	<i>X</i>
$a + b$	\rightarrow	$c + d + e,$	<i>XI</i>
a	\rightarrow	$b + c + c,$	<i>XII</i>
a	\rightarrow	$b + c + d + d,$	<i>XIII</i>
a	\rightarrow	$b + c + d + e,$	<i>XIV</i>
$a + a$	\rightarrow	$b + c + c,$	<i>XV</i>
$a + a$	\rightarrow	$b + c + d,$	<i>XVI</i>

where a , b , c , d , and e represent different species. M represents any other species. Specific coding is written for each type of reaction. This allows the fastest possible coding to be used for each. Each reaction listed actually includes both the forward and reverse reactions.

As pointed out earlier, each reaction rate is calculated as the product of a temperature-dependent term times the appropriate reactant concentrations, i.e.

$$R_k = f_k(T) \prod_i c_i, \quad (7.1)$$

where

$$f_k(T) = A_k T^{b_k} \exp(-E_k^a/RT). \quad (7.2)$$

The terms A_k , b_k and E_k^a , as well as the indices of the species going into and coming out of the reaction are stored in the disk file CDAT.

From Eq. (2.1) we see that these reaction rates are evaluated implicitly in terms of T_j' and $c_{i,j}'$, the values at the end of the computational cycle. This choice prevents $c_{i,j}'$ from being negative. This is true of all quantities except

$$c_j^{tot} = \sum_i c_{i,j}, \quad (7.3)$$

which corresponds to M in reaction types III and IV. c_j^{tot} is evaluated explicitly to simplify the coding, since there is no possibility that a species will become negative as a result of this choice.

It sometimes happens that one will be using a rate from CDAT in a temperature range where the coefficients A_k , b_k or E_k^a are inappropriate. This often happens for the activation energy E_k^a of a reverse rate. These may be negative in order to have the correct equilibrium with its forward rate in a given temperature range. Since the rate depends exponentially on E_k^a , it can be shown to result in rates much too large outside the temperature range for which it was derived. This can cause numerical difficulties since a calculation will require more cycles to follow the rapid changes in species affected by this rate.

The code allows a maximum rate \mathcal{R}_{max} to be specified and then uses

$$\mathcal{R}_k^{eff} = \frac{\mathcal{R}_{max} \mathcal{R}_k}{\mathcal{R}_{max} + \mathcal{R}_k} \quad (7.4)$$

instead of \mathcal{R}_k in the actual calculation

For two-body reactions the code uses

$$\mathcal{R}_{max} = k_1 c_a c_b, \quad (7.5)$$

where k_1 is an input constant. For three-body reactions involving M , c_a and c_b we also use

$$\mathcal{R}_{max} = k_1 c_a c_b. \quad (7.6)$$

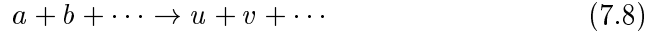
In this case we argue that a three-body reaction is limited by its slowest two-body partial reaction.

The forward parts of reaction types V and VI represent spontaneous decays. The code uses

$$\mathcal{R}_{max} = \frac{1}{\tau_1} c_a, \quad (7.7)$$

where τ_1 is an input constant.

In addition to the specific forms I-XVI, the code allows the user to specify arbitrary reactions of the form



where either

$$\mathcal{R}_k = f_k(T) \prod_m c_m^{\alpha_m}, \quad (7.9)$$

or

$$\mathcal{R}_k = f_k(T) c_{tot} c_m^{\alpha_m}.$$

The c_m in Eq. (7.9) need not be the c_i appearing in Eq. (7.8), and are raised to the α_m power in calculating R_k . $f_k(T)$ still has the same form as in Eq. (7.2). The rate-limiting combination for \mathcal{R}_k^{eff} is not used.

Boundary Calculations

The code carries an extra dummy zone at each end of the grid to handle boundary conditions. The values assigned to the variables describing the dummy zone are set so that the desired boundary conditions are automatically satisfied. Thus there exists a relation of the form

$$\vec{y}_B = \vec{g}(\vec{y}_J), \quad (8.1)$$

where \vec{y}_B is the vector of boundary variables and \vec{y}_J is the vector of the neighboring real zone variables. Thus

$$\delta \vec{y}_B = \delta \vec{y} \frac{\partial \vec{g}}{\partial \vec{y}_J} \quad (8.2)$$

and for the last real zone Eq. (4.4) becomes

$$\vec{f}_J^n + \delta \vec{y}_J \left\{ \frac{\partial g}{\partial \vec{y}_J} \frac{\partial}{\partial \vec{y}_B} + \frac{\partial}{\partial \vec{y}_J} \right\} \vec{f}_J^n + \delta \vec{y}_{J\pm 1}^n \frac{\partial \vec{f}_J^n}{\partial \vec{y}_{J\pm 1}} = 0. \quad (8.3)$$

Thus the equation for zone J couples only its own variables with its one neighboring real zone.

The general procedure is that at the beginning of the calculational cycle the boundary variables are set using Eq. (8.1). Then the vector \vec{F}^n and T^n (if required) of Eq. (4.5) are calculated for the real zones next to the boundary just as for interior zones. If T^n is calculated, a subroutine is called that combines the coefficients of $\delta \vec{y}_B$ with the coefficients of $\delta \vec{y}_J$ as implied by Eq. (8.3). For a fixed boundary Eq. (8.1) becomes

$$\begin{aligned} c_{i,B} &= c_{i,J} \\ T_B &= T_J \\ v_B &= 0 \end{aligned} \quad (8.4)$$

The first two relations insure that the pressure gradient across that boundary is zero, and thus the boundary will not accelerate. We also insure that nothing diffuses across the boundary by choosing

$$\begin{aligned} (D_i)_B &= 0 \\ (\alpha^T)_B &= 0 \end{aligned} \quad (8.5)$$

although this is redundant if the gradients are zero.

Recall that the vector \vec{x}_j contains the variable $v_{j-\frac{1}{2}}$. Thus at the lower boundary solving Eq. (4.5) solves an equation for v_B . At $j = j_{max}$ the v associated with $\vec{x}_{j_{max}}$ is between zones j_{max} and $j_{max} - 1$, not at the boundary. At $j = j_{max}$ v_B is obtained directly from Eq. (8.1).

At $j = 1$ setting $v_B = 0$ gives troubles in defining $\delta v_B / v_B$. In this case we substitute the equation

$$v_B = 0 \quad (8.6)$$

for Eq. (2.9) after the elements of \vec{F}^n and T^n have been filled in the usual way.

For an open boundary we assume that the dummy zone is at a pressure P_B and choose

$$\begin{aligned} c_{i,B} &= \alpha c_{i,J} \\ T_B &= T_J \end{aligned} \quad (8.7)$$

where α is a proportionality coefficient chosen so that

$$P_B = \sum_i c_{i,B} R T_B. \quad (8.8)$$

Clearly,

$$\alpha = \frac{P_B}{P_J} \quad (8.9)$$

Equations (8.7) insure that there is no diffusion of species or thermal conduction into the dummy zone. This is also guaranteed by enforcing Eqs. (8.5).

Equation (2.9) then gives the velocity at the boundary, where the viscosity Q is taken to be zero in the dummy zone.

If a calculation is dynamically adjusting its grid size, we use a different T_B in the rezoning algorithm. If $T_B = T_J$ the rezoning algorithm would make Δr_J too large, because it would assume no gradient between the dummy zone and the first real zone. Instead we fit the temperature in the first two real zones to a second-order polynomial in x that has zero slope at the boundary. We then choose T_B so that Eq. (6.29) gives the polynomial's value at the boundary. This gives

$$T_B = T_J + \frac{2}{\alpha(2 + \alpha)} (T_J - T_{J\pm 1}), \quad (8.10)$$

where

$$\alpha = \frac{\Delta r_J + \Delta r_{J\pm 1}}{\Delta r_J} \tag{8.11}$$

Energy Sources

To add energy to a problem, HCT has a separate routine that calculates a source deposition rate for each zone. This source rate can be specified as either a source per unit volume or a source per unit mass. The routine adds the energy as an additional term on the right side of Eq. (2.13). If the rate is given per unit volume, the additional term is

$$\Delta \mathcal{E}_j^s = \Delta t \int_{\Delta r_j} dx S^{vol}. \quad (9.1)$$

If the rate is given per unit mass, the appropriate term is

$$\Delta \mathcal{E}_j^s = \Delta t \rho_j \int_{\Delta r_j} dx S^{mass}. \quad (9.2)$$

Note that for simplicity either term is evaluated explicitly in terms of the quantities at the beginning of the cycle.

The effect on the corresponding differential equations is to add either S^{vol} or ρS^{mass} to the right side of Eq. (3.14). These terms are divided by C_V if added to the right side of Eq. (3.16).

The integrals in Eqs. (9.1) and (9.2) are done exactly, assuming that S^s is distributed in space according to

$$S^s = S_0^s \frac{1}{1 + \exp\left(\frac{x - x_0}{\Delta x}\right)} \quad (9.3)$$

where S_0^s is a constant. The superscript s refers to either a source per unit volume or a source per unit mass. The integrals are done exactly so the step in Eq. (9.3) will be accurately taken into account even if Δr_j is much larger than Δx .

The form chosen for Eq. (9.3) allows one to specify a source that is nearly constant from $x = 0$ to $x = x_0$, with a decay to 0 over a distance Δx . A

point source at $x = 0$ is specified by choosing $x_0 = 0$ and making Δx small. Δx is usually chosen large enough to keep the source from changing abruptly from one zone to the next.

The integrals in Eqs. (9.1) and (9.2) are evaluated using

$$I(x) = \int dx \frac{1}{1 + e^y} = x - \Delta x \ln(1 + e^y), \quad (9.4)$$

$$y = \frac{x - x_0}{\Delta x}$$

Then

$$\int_{\Delta r_j} dx \frac{1}{1 + e^y} = I(x_{j+\frac{1}{2}}) - I(x_{j-\frac{1}{2}}).$$

Since y appears in the exponential, evaluating Eq. (9.4) can cause machine overflow when y is large. This often happens, since y varies inversely with Δx . To avoid this, we use the first term of the asymptotic expansion

$$I(x) = x_0 - \Delta x (e^{-y} + \dots), \quad (9.5)$$

when x is a large positive number. Similarly, if x is large but negative, we use the first term of

$$I(x) = x - \Delta x (e^{-|y|} + \dots). \quad (9.6)$$

Currently asymptotic values are taken if $|y| > 100$.

Using Eqs. (9.4), (9.5) we get

$$\int_0^\infty dx \frac{1}{1 + \exp\left(\frac{x - x_0}{\Delta x}\right)} = x_0 + \Delta x \ln \left\{ 1 + \exp\left(-\frac{x_0}{\Delta x}\right) \right\}. \quad (9.7)$$

This allows one to calculate the total energy source rate if Eq. (9.1) is used. The source rate can be applied from time $t = 0$ up to some specified time $t = t_{ig}$ in the current version of the code.

If the calculation is dynamically adjusting its zone size when the source is being applied, we no longer assume in the rezoning algorithm that the

temperature at the problem boundary has zero slope. Rather, we linearly extrapolate the temperature in the first two real zones to the boundary, and choose T_B so that Eq. (6.29) is consistent with this value. Equation (8.10) is replaced with

$$T_B = T_1 + \frac{2}{\alpha} (T_1 - T_2) \quad (9.8)$$

at the $x = 0$ boundary.

Calculational Cycle

In this section we describe how the code that solves the difference equations is used to solve a particular physical problem.

The underlying principle is to have the main code act as a black box which, given the parameters describing a particular situation, integrates the equations of motion and provides edits of the problem as it is running. Thus initially the main code itself does not contain any information about a particular problem. The data describing a problem is read from an input file on disk. The input file contains all the parameters necessary to describe the physical problem, plus parameters that indicate how the problem is to be run. It also contains lists of species and reactions to be used. Appendix A gives a complete glossary of the variables that may be set in the input file. The data to describe the physical constants associated with each species and each reaction exists in a separate disk file CDAT.

The input file also indicates how frequently to edit the problem. These edits are the only way that the code produces output for the user. There are currently six edits, each giving a different kind of information about the problem. These edits are called independently of each other.

The first is an edit of general zone quantities. Position, size, velocity, total species concentration, density, temperature, pressure, and rates of energy production are printed for each zone.

A second edit provides a list of the mole fraction of each species in each zone of the problem.

A third edit gives a (complete or partial) listing of the rates of reactions in (some or all of) the calculational zones.

A fourth edit provides a list of the total relative change of each physical variable in that timestep.

The above four edits are Lagrange edits, i.e., they provide a picture of how the problem is changing in the Lagrange part of the calculation. Calling the fourth edit automatically gives an edit of the total change due to the

variable remap if an Eulerian calculation is being done. This gives one a method for estimating the change at a point due to advection.

A fifth edit is used for debugging purposes. It gives a listing of how the variables change during the iterations in the solution of the nonlinear equations of motion.

The sixth edit acts as a dump routine. This routine writes all the physical variables into a binary disk file at selected cycles. These dumps, together with the original input deck, contain sufficient information to restart the problem at one of the selected cycles. The dumps also contain sufficient data for input to a graphics program that is used in a postprocessor mode.

At the start of a problem, the main code calls the subroutine SETUP. SETUP reads an input deck and echoes it to the printer. Then SETUP calls the routine EOSDAT, which reads the data for the appropriate species and reactions from the disk file CDAT and stores it in the code's common blocks. SETUP may also call routines to initialize the data in a way determined from the input deck. If the input calls for restarting from the dump of a previous run, SETUP calls the routine RDUMP to read the initial values from the dump file.

The main code then loops through the following sequence of operations for each calculational cycle:

First it checks to see if edits are called during this cycle. If they are, it sets flags that will cause the appropriate edit routines to be called during the cycle.

Next, the main code calls CALCOEF. This routine calculates all the coefficients that are held fixed during the cycle, such as the energy source terms and the diffusion coefficients. If the energy source is on, the code also calls another routine SORS. SORS can turn off the source when the energy production rate from chemistry has reached a predetermined level. This allows one to apply a source only as long as the chemistry is not producing significant energy.

Next, the physical variables are extrapolated from the cycle start time to the cycle end time using Eq. (4.14). This provides the initial guess for the final values of the physical variables.

We are now ready to evaluate the terms in Eq. (4.8). If the matrix T^m is not to be updated for this iteration, subroutine DIFFUN is called. DIFFUN only evaluates \vec{F}^n . If the matrix T^n is also to be calculated, DIFPED is called. This routine evaluates the terms in both T^n and \vec{F}^n . For ease of coding, DIFFUN and DIFPED are divided into sections in which only one aspect of the physics is updated at a time. Thus the equations with no interaction terms are evaluated first. Next the hydro terms are added, followed by the thermal-conduction terms. The species-diffusion terms are added next and then finally the chemistry terms. Separating the coding for each type of physics makes it relatively easy to add new physics to the code, or to debug the coding associated with any physical process.

If a new T^n has been calculated, DECBTL is first called to do the LU decomposition given by Eq. (4.7). In either case, SOLBTL is then called to solve Eq. (4.8).

After the linearized equations have been solved, CKCNVG is called to see if the system has converged to the solution of the nonlinear equations. The system has converged if Eqs. (4.9) and (4.10) are satisfied. CKCNVG also decides whether a new evaluation of T^m is needed for the next iteration.

If an edit of the results of the iteration is desired, the iteration edit routine is called. If the iteration has not converged and the number of iterations has not exceeded a preset maximum, a new iteration is tried. If the preset maximum number of iterations has been reached, the timestep is cut in half and the whole cycle started over again, beginning with a new extrapolation of the physical variables to the new end time. This cutting of the timestep and retrying the cycle is allowed only a (second) preset maximum number of times. If the iterations have not converged after this number of successively smaller timesteps, the problem is terminated. If the iterations have converged, we now have the physical variables of the Lagrange calculation at the cycle end time.

If the problem is Eulerian, the rezoning routine is called to determine the new grid. For both Eulerian and Lagrangian calculations the Newton polynomials containing the past history of the physical variables are then updated.

Next, a new timestep Δt is calculated for the next cycle. Our criterion for a timestep is that the maximum relative change during the Lagrange

calculation of the physical variables be limited to a fixed number, typically 0.05 or 0.1. Species concentrations below a cutoff value are allowed a larger relative change, since it is believed that they do not need to be accurately determined to give an accurate solution to the important variables. The criterion for them is to keep them small enough so that the expansion to first order in small quantities is still accurate enough for the system to converge.

After the zone mass and zone energy are updated, the code calls the Lagrange edit routines if the appropriate edit flags have been set. If the calculation is Eulerian the remap routine is called next. Then, if appropriate, the dump routine is called. All the physical variables have now been advanced from time t to time $t + \Delta t$, and the code is ready for the next calculational cycle.

Some Programming Details

The following section is intended for those who might be looking at the actual coding of HCT. We try to give an idea of the thinking behind the organization of the coding itself, rather than the mathematics involved in solving the physical problem.

To allow one to calculate arbitrarily sized problems, the coding for HCT is set up with a modified form of dynamic dimensioning. Arrays are dimensioned at compile time to be large enough to contain the largest problem one reasonably expects to run. Any smaller problem then uses only a fraction of the total space reserved. To do this efficiently, all logically two- and three-dimensional arrays in these dimensions are laid out in memory as one-dimensional arrays.

The two-dimensional arrays are stored one column after another continuously in memory, with the declared dimension of the array limiting only the total length of the associated data. To do the bookkeeping for a particular calculation, the code determines from the input deck the following integers associated with a given problem:

KMX is the total number of reactions in a problem, including both those calculated using Eq. (7.1) and Eq. (7.9). The number of special reactions using Eq. (7.9) is KMXS. These are assigned indices K less than those assigned to the standard reactions using Eq. (7.1). Thus a loop over all reactions is written in two parts. The first part calculates reactions with the form of Eq. (7.9), the second part calculates reactions with the form of Eq. (7.1).

JMX is the number of zones in the problem, including the dummy boundary zones. Then JMXM2 = JMX - 2 is the number of real zones.

IMX is the number of species in the problem.

IT = IMX + 1 is the index of the zone temperature. The zone variables are arranged as in \vec{x}_j (see discussion of the matrix formulation of the solution procedure). The temperature follows the species in memory.

IR = IMX + 2 is the index of the zone width.

$IU = IMX + 3$ is the index of the velocity of the lower boundary of the zone. Since this is the last zonal variable, IU is also the number of variables associated with a zone.

The code is dimensioned for a maximum $MXJMX$ zones and $MXIU$ zone variables. A problem with less is allowed since JMX and IU are calculated for each problem. The two-dimensional arrays such as the vector \vec{X} are dimensioned $MXIUJMX$. This limits the maximum number of species times zones. Any problem for which the product of IU and JMX is less than $MXIUJMX$ can be run without recompiling the code.

For example, the vector \vec{X} , which can be thought of as a two-dimensional array $C(I,J)$ (including T_j , Δr_j and $v_{j-\frac{1}{2}}$ as generalized $c_{i,j}$), is stored as a one-dimensional array $C(N)$. To pick out the i^{th} species in zone j , HCT has coding explicitly written to fetch $C(N)$, with

$$N = I + (J - 1) * IU \quad (11.1)$$

The arrays A_j , B_j and C_j in the matrix T of Eq. (4.5) are IU by IU . Elements like $(A_j)_{q,r}$ of A_j involve terms like $\partial(\vec{f}_j)_q / \partial(\vec{x}_j)_r$. The location, or offset, within the space allocated for A_j is given by

$$N = q + (IU - 1) * r \quad (11.2)$$

The A_j 's are then stored one after another by zone.

One notes that if $r = IT$, IR or IU we can save a multiply at fetch time by calculating

$$\begin{aligned} IIT &= (IU - 1) * IT \\ IIR &= (IU - 1) * IR \\ IIU &= (IU - 1) * IU \end{aligned} \quad (11.3)$$

at problem generation time. We also precalculate the offsets for (q, r) pairs that have both q and r either IT , IR or IU :

$$\begin{aligned}
ITIT &= IT + IIT \\
IRIT &= IR + IIT \\
IUIT &= IU + IIT \\
ITIR &= IT + IIR \\
IRIR &= IR + IIR \\
IUIR &= IU + IIR \\
ITIU &= IT + IIU \\
IRIU &= IR + IIU \\
IUIU &= IU + IIU
\end{aligned} \tag{11.4}$$

HCT is currently implemented on the CDC 7600's at Lawrence Livermore Laboratory. At LLNL the 7600 has a two-level memory—a 50,000+ word fast access small-core memory (SCM), and a 400,000+ slower access large-core memory (LCM). A typical large problem that we have run has IU around 20, with about 25 zones. Thus the matrices A_j , B_j and C_j that make up the matrix T are on the order of 400 words each. The memory necessary to save the whole matrix T is on the order of 30,000 words. This is too large to fit in SCM and still allow room for the rest of the code, so it is stored in LCM. The IU-by-IU arrays A_j , B_j and C_j are block copied in and out of SCM so that the calculations can be done when the data are available in the faster access memory.

This means that one has access to only one zone at a time when the matrix T is being calculated. Thus the calculations have been organized on a zone-by-zone basis. A double loop running over a variable with both species components and zone components will start by calculating all the species components for the first zone, and then increment the zone counter. We generally adhere to this convention even if all the data are available in SCM at a given time.

Another important point about the programming is that we have been able to use highly optimized assembly language routines in the matrix manipulation parts of the code. These routines take advantage of the “pipeline architecture” of the 7600 in a way that FORTRAN coded routines cannot.

As pointed out before, the number of operations associated with doing the LU decomposition of our matrices increases roughly as the cube of the

number of species. At somewhere around 15 to 20 species, even using these optimized routines, the time spent doing the matrix inversions becomes comparable to the time spent in the rest of the code. Clearly, if one expects to do relatively large calculations, the time spent optimizing these routines is time well spent.

These matrix-manipulation routines include DEC, which does the LU decomposition of a general matrix, and SOL, which does the back-substitution solution of a system of linear equations once the LU decomposition has been done. In carrying out the matrix operations in Eqs. (4.7), we use MMM, which multiplies two matrices together and adds the product to a third, and QVCOMP, which changes the sign of every element in a matrix. The back-substitution algorithm also uses MMV, which multiplies a vector by a matrix and adds that to a third vector. These routines are in the STACKLIBE[8] library, available at Lawrence Livermore Laboratory. Other installations may wish to obtain these routines, or write their own equivalent routines.

The rest of the code is written in LRLTRAN,[9] an extended FORTRAN. This allows the code to be easily changed, as well as more easily understood. As the above discussion indicates, assembly language coding of the physics would not have a large payoff in terms of efficiency in large calculations, since so much of the calculational time is spent in the matrix algebra. A complete listing of the FORTRAN coding is given in Appendix B.

The graphics output is done with a postprocessor. The postprocessor reads dumps made by HCT when the problem was run. It uses the Lawrence Livermore Laboratory DISPLAY[10] package to produce hardcopy. DISPLAY is capable of producing hardcopy graphs and 16-mm movies in either color or black and white. Having the graphics produced by a postprocessor rather than inline allows one to change the form of the graphics without rerunning the original problem.

Since DISPLAY uses hardware and software that may be unique to Lawrence Livermore Laboratory, users at other installations will probably have to modify the graphics routines.

Examples

In this section we consider several sample problems that illustrate capabilities of HCT. These sample problems are not to be thought of as “optimized” descriptions of physical problems. No attempt has been made to verify the correctness of the physical parameters used. The parameters are chosen to be correct to roughly an order of magnitude. The purpose of these examples is to indicate the nature of the calculations the code can handle, and the kind of information the edits are designed to provide. We also discuss the numerical characteristics of the code as implemented on the CDC 7600 computers at Lawrence Livermore National Laboratory.

The first problem we consider involves only hydrodynamics and thermal conduction. It is designed to illustrate the behavior of implicit hydrodynamics when quantities are changing slowly compared to sound transit times. Figure A.0 shows the problem after one cycle. It consists of 1.0 cm of nitrogen at one atmosphere (taken as $1.0 \times 10^6 \text{ dynes/cm}^2$). The left 0.4 cm is at 2400 K, the right 0.6 cm at 300 K. The discontinuity has the form of Eq. (9.3), with Δx equal to 0.01 cm. The thermal conduction coefficient is given by

$$\alpha^T = \left(4.58 \times 10^{-7}\right) \frac{T^{\frac{1}{2}}}{c_{tot}} \quad (12.1)$$

The problem has 25 spatial zones, with N^* , the number of zones reserved to define the temperature gradient, equal to 5. This results in a minimum zone size of 0.0081 *cm* and a maximum of 0.098 *cm*. The code was allowed a 0.1 relative change in the physical variables each cycle.

Since there is only one species in the problem, the mole fraction plot is of no interest. Since the figure was made after one cycle, the velocities have changed from their initial zero value.

Figure A.1 shows the state of the problem at 1.16 *msec*. Heat has diffused from the warmer region, lowering its temperature and raising its density as the cooler material moves to the left. Note that the timestep is $1 \times 10^{-4} \text{ sec}$, with the smallest zone $2.6 \times 10^{-2} \text{ cm}$ wide. The sound speed at 1400K is about $7.5 \times 10^4 \text{ cm/sec}$, giving a sound transit time across the smallest

zone of $3.5 \times 10^{-7} \text{ sec}$. If this were an explicit calculation, we would need a timestep of this order. However, since the hydro calculation is implicit, we are using timesteps several orders of magnitude greater than the Courant limit. As discussed before, the hydrodynamics is not describing any sound waves that might be present. It is only moving material around so that pressure equilibrium is established.

One notices that the minimum temperature appears to be slightly below $300K$ at the bottom of the high temperature gradient region. In fact, it is $292K$ there, with a slight hump as one moves to the right. This indicates that the continuous rezoning is introducing at least a 3 percent error there. The total internal energy in the problem was conserved to within seven significant figures, indicating that the numerical approximations have slightly distorted the temperature profile while conserving energy.

The problem was rerun to this time with 50 zones and an N^* of 10. The minimum temperature then increased to over $300K$, although there was still a slight hump in the fourth significant figure. In general one will always have to trade off accuracy and cost in such a way as to be confident of the results while still having an affordable calculation.

The problem was run to a time of 1.0 second. At that time the temperature ranged from 503 to 521 K . The timestep had climbed to $5 \times 10^{-2} \text{ seconds}$. During the calculation the factor limiting the timestep was the requirement that the boundary of a zone not move through more than 0.1 of a zone.

Assuming that each cycle represents a significant change in the physical variables, questions of computer costs divide naturally into four main parts. The first point of interest is the cost of evaluating the left and right sides of Eq. (4.5). The second is the cost of solving Eq. (4.8). This depends on whether the LU decomposition of T^m has already been done or not. Since the LU decomposition is considerably more expensive than the solution by back substitution, the third important question is the frequency of LU decompositions compared to the total number of iterations. And finally, since the cost is proportional to the number of iterations, the final point of interest is the average number of iterations per cycle.

Since all computer costs scale by the number of zones and number of subroutine calls, we generally quote timing results for calls in terms of their cost per zone calculation. This form provides conveniently normalized reference

numbers. Costs for other calculations can then be estimated by knowing how the individual routines scale. For example, DIFFUN and DIFPED will usually scale by the number of reactions for problems with large numbers of reactions. The cost of the hydro is usually small compared to the cost of evaluating many reactions. However, the cost of the LU decompositions asymptotically scales by the cube of the number of species and depends not at all on the number of reactions. Thus if the matrix manipulation routines are the dominant sink of CPU time, one knows that adding more reactions will not increase costs much if no additional species are added.

For this problem, the cost of evaluating the left side of Eq. (4.5) averaged $46 \mu\text{sec}/\text{zone} \cdot \text{call}$. Calculating both the left and right sides of Eq. (4.5) took $126 \mu\text{sec}/\text{zone} \cdot \text{call}$. Each LU decomposition required $219 \mu\text{sec}/\text{zone} \cdot \text{call}$, and solving the system by back substitution required $66 \mu\text{sec}/\text{zone} \cdot \text{call}$. This problem represents the minimum cost for the matrix-manipulation routines, since it has the minimum 4 variables/zone. Since there is no chemistry, the costs of evaluating Eq. (4.5) are also minimized.

The code averaged 5 iterations/cycle to converge to 1 part in 10^6 . New LU decompositions were done 110 times in 175 cycles, averaging 1 call per 8 iterations

The entire calculation averaged about one $\text{msec}/\text{zone} \cdot \text{cycle}$. Thus to run the 25-zone problem to the 50 cycles of Fig. 2 took approximately 1.3 seconds of CPU time.

These figures are given to indicate the cost of the calculations on the CDC 7600. They are very much affected by the relative efficiency of the coding. In particular, it is important to recall that the matrix manipulations are done by calls to assembly-language routines that are probably nearly as efficient as possible on this machine.

The next example is a simple ozone flame. Table A.1 gives the constants associated with the equations of state for O , O_2 and O_3 used in the calculation. CPA1, CPA2, etc. are the coefficients of the specific heats expanded in powers of $T/1000$, with CPA1 being the coefficient of $(T/1000)^0$. The units of C_p are cal/mole . H0 is the enthalpy at $298K$, in kcal/mole .

Table A.2 gives the constants associated with the reaction set used. FRC, FRD and FRX correspond to A_k , b_k and E_k^a in Eq. (2.2) for the forward

reactions. RRC, RRP and RRX are the corresponding terms for the reverse reactions. IRF is the reaction type, according to the six possible types explained in the kinetics description.

In this problem we mimic Spalding's approach to finding the flame velocity. Spalding's approach is to choose an initial configuration and evolve the time-dependent equations of motion until steady state is achieved.[11] Our procedure differs slightly in that we solve the full hydrodynamic equations rather than assume pressure equilibrium. To force our equations to give pressure equilibrium and hasten the approach to steady state, we reduce the acceleration term Dv/Dt in Eq. (3.8) by a factor of 10^3 . This reduces the inertial resistance to the establishment of pressure equilibrium.

The initial condition is shown in Fig. A.2. The mesh is 0.3 cm long. The left 0.02 cm is at 1250K and the right 0.28 cm at 300K. The unburned mixture consists of a 1:3 mole fraction mixture of $O_3 : O_2$, with the O mole fraction at 10^{-6} . The burned gases were given initial mole fractions of 0.01, 0.99 and 2×10^{-5} for O , O_2 and O_3 , respectively. The burned-gases temperature of 1250K is approximately the adiabatic flame temperature of the mixture. The discontinuity between the left side and the right side was again chosen to have the form of Eq. (9.3), with $\Delta x = 0.001cm$. As in the first example, there are 25 zones in the problem, with 5 zones reserved to describe the temperature gradient. The system is at one atmosphere, with the left boundary closed and the right boundary open.

The thermal conduction coefficient is given by Eq. (12.1). The species diffusion coefficient by

$$D_i = (4.16 \times 10^{-7}) \frac{T^{\frac{1}{2}}}{c_{tot}} \quad (12.2)$$

the same for each species.

The problem was started with an initial timestep of $10^{-9}sec$ to allow the fast kinetic rates to come into equilibrium. Figure A.3 shows the problem after $15.7\mu sec$ at cycle 100. The problem has not yet settled into a steady state, since the velocity is negative behind the flame front. This indicates that the whole system is being pushed to the left. The step in temperature has rounded somewhat.

Figure A.4 shows the flame when it has propagated most of the way across the grid at $389\mu sec$ and cycle 1000. The flame now is nearly at a

steady state, but there are still effects present due to the fixed left boundary. From the mole-fraction plot, one notes that the O_3 concentration is still decreasing as one approaches the left boundary. Later we will show that heat is being produced behind the flame. This results in heat flowing to the right. Since there is no heat flow through the left boundary, the boundary is unlike the corresponding point in the infinite-medium case. As the flame gets farther from the left boundary, the characteristics of the flame approach those of the infinite-medium case.

Table A.3 shows an edit of the problem at $3.89 \mu sec$. The left column labelled JD gives the zone number. The column labelled R gives the location of the left boundary of the zone relative to the left boundary of the problem. DR is the zone width, CTOT the total molar species concentration, RHO the density, T the temperature and U the velocity of the left zone boundary. PRESS is the pressure in CGS units.

EDOTR is the energy produced through chemical reactions in the zone. EDOTC is the energy deposited by conduction, EDOTD that deposited by the species diffusion. $P * DV/DX$ is the work produced by hydrodynamics. Each of these terms corresponds to a term in Eq. (3.16). EDOTC is the term proportional to α^T , EDOTD the term proportional to the D_i .

VSOUND is the velocity of sound in the zone. From VSOUND and the fact that the timestep is around $4 \times 10^{-7} sec$, we note that the problem is running more than an order of magnitude above the Courant condition.

The other parameters shown at the top of the edit relate to the way the problem is running. AVGITER is the average number of iterations/cycle. NJCBTOT is the number of times the full LU decomposition of the equations was done. IRCYTOT is the total number of times the iteration scheme failed to converge. IRZFTOT is the number of times the rezoning routine tried to move the grid more than allowed. The code has a parameter that sets the maximum distance the rezoning routine can move a zone boundary. That parameter for this problem is 0.1 times the width of the zone in the direction of the move. If the desired distance is greater than allowed, the rezoner moves the boundary as far as allowed, adds one to IRZFTOT, and the calculation proceeds. If the rezoner continually tries to move boundaries farther than allowed, one may want to run the problem over with the timestep reduced.

At the bottom of the edit is information about how much computer time has been used so far. A line giving the total CPU time, I-O time and system-call time is followed by a table giving the time in microseconds for various subroutines. If the values in the second column are divided by the values in the first, we get the number of calls to the subroutines each cycle. SOL and DEC terms are for the full block-tridiagonal-matrix routines.

From EDOTR in Table A.3 one sees that zone 12 is the zone with the largest heat release due to kinetics. Table A.4 is an edit of the rates of change of the species in this zone.

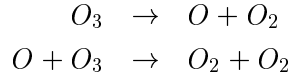
The first two rows identify the physical variables and give their values. The next two rows give the total and net rates of change of the variables in $\text{moles}/\text{cm}^3 \cdot \text{sec}$ (except for temperature, which is given in K/sec). The total rate is the sum of the absolute value of the rates of change due to conduction, diffusion, hydrodynamics, and each forward and reverse chemical rate. Thus it is a measure of the shortest characteristic timescale, or stiffness, of that component of the system. The net rate is just the sum of the rates, and thus the rate at which the variable is changing. By looking at these two numbers one can tell if the variable is stiff or not, as discussed in connection with the characteristics of chemical systems.

The numbers under each physical variable following the total and net rates are expressed as fractions of the total rate. Thus one can see at a glance which rates are controlling a given variable. The actual value of the rate of relative hydrodynamic change and the actual rates of the chemical reactions are given under the RATE column.

The hydrodynamic rate of change of a species concentration c_i is just $c_i * (-dv/dx)$, where v is the velocity. It is due to the expansion or contraction of the zone. The relative rate is $-dv/dx$. This is a Lagrange edit, taken after the Lagrange part of the cycle, so the advection implied by the remap is not taken into account.

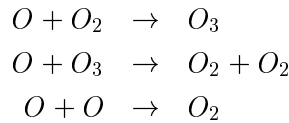
One notes that there are only two reactions printed. The code does not print out rates that are smaller than a given fraction of the total rate for each species involved. This cuts down the size of the edit when many reactions are being calculated but few are important. In this edit that fraction is 0.01. Thus the remaining rates in Table A.4 give rise to contributions smaller than 1 percent of any species' total rate. There is also an option to set this fraction to zero, and thus print out all the rates.

Examination of Table A.4 shows that the chemistry is proceeding via



These rates are not quite equal because of the net diffusion of O into the zone. The O itself is in a quasi-steady state, since the net rate is very much smaller than the total rate. One notes a significant amount of diffusion of O_3 into the zone and a significant diffusion of O_2 out of the zone.

Now let us look at the region behind the flame, where slow burning is causing the boundary effect to be large. We look at zone 4, which is more than half way back to the left boundary. Table A.5 shows that the important chemistry is



A negative reaction number in the table indicates a reverse reaction. The first two reactions are nearly balanced, and the net effect is the same as the third reaction. The third reaction rate is comparable to the net rate of the first two, and all the reactions are exothermic. Thus at this point we are converting excess O into O_2 with a release of heat. Note that here the O_3 is the stiff component.

Next let us investigate how quickly the problem is approaching the steady-state infinite-medium conditions we expect as the flame front gets farther and farther from the left boundary. A measure of approach to steady state is the rate of change of the flame velocity.

We use the definition that the flame speed is the speed of the flame front relative to the unburned gas. We define the flame front to be the point at which the temperature is a predetermined value T_f . It is usually advantageous to pick a T_f that is characteristic of the steepest portion of the temperature curve, since the automatic rezoner gives the best resolution there. Even so, since the problem is divided into discrete zones, the temperature variation is not known explicitly across any one zone. The best that can be done is to assume an interpolation based on the average value of the temperature in the zone.

The prescription we use is to assume that T_j represents the temperature at the center of the zone. We linearly interpolate from zone centers to define the temperature $T(x)$ everywhere. The location where $T(x)$ assumes a particular value can then be found.

We can be most confident that we accurately know when a flame has moved a distance Δx in time Δt if, during that time, the flame has moved from a location that is a fraction α across one zone to a location that is the same fraction α in another zone. Then the inaccuracies inherent in the interpolation are minimized.

One might think that the distance Δx characteristic of the distance the flame front moves when it moves through a single calculational zone would be Δr_{min} , the size of the smallest zone. When using the dynamic rezoning option, however, this is not true, since the grid tends to move with the flame front. In fact, Δx turns out to be more like Δr_{max} , the size of the largest zone.

One can understand this result by picturing a steady state flame in the middle of the grid. The size of a zone in the flame front is determined by the requirement that it represent no more than a specified fraction of the temperature change across the grid. The zones become larger as one moves away from the flame front so that the entire grid is described. Now imagine the flame has moved the distance of one of the larger zones. The number of zones needed to describe the front is the same, but now we need one less zone ahead of the front and one more behind it to describe the whole problem. Thus the flame must have burned through one calculational zone.

The effect of this change is the same as if one of the larger zones had been moved from ahead of the flame to behind it. The actual size of the zone moved will vary as the flame moves across the grid because the algorithm determining the grid is nonlinear. One can see this in Figs. A.2, A.3, and A.4.

We measure the velocity by measuring the movement of the front between problem dumps. For each dump n we determine the location of the flame front within the zone containing T_f . Then we locate the next dump m with T_f nearest the corresponding location in the next zone. Thus

$$v_n(\Delta x) = \frac{x_m - x_n}{t_m - t_n} \quad (12.3)$$

We write $v_n(\Delta x)$ to emphasize that the velocity is measured over a finite distance. As discussed above, that distance may not be the same at different times in the same problem even if the problem is in a steady state. The time to propagate through a zone depends on the location of the flame, since Δr_{max} varies as the flame moves across the grid.

The time associated with the velocity $v_n(\Delta x)$ is given by

$$t_n(\Delta x) = \frac{1}{2}(t_m + t_n). \quad (12.4)$$

One may have dumps close enough in time that $m > n + 1$. In that case there are several $v_l(\Delta x)$ that represent velocity measurements starting between x_n and $x_n + \Delta x$. Since we are already averaging over a distance Δx , we can average over these measurements to improve resolution and still be sure that we are measuring properties only over a distance characterized by Δx . Thus let

$$v_n(\Delta x) = \frac{1}{m - n + 1} \sum_{l=n}^m v_l(\Delta x) \quad (12.5)$$

$$t_n(\Delta x) = \frac{1}{m - n + 1} \sum_{l=n}^m t_l(\Delta x) \quad (12.6)$$

In addition, one can improve resolution by measuring velocities over Δx 's that correspond to the flame moving through more than one zone, if desired.

An important question that has yet to be addressed is how quickly the numerical procedure converges to the solution of the differential equations. We use the calculated flame velocity to measure this.

Figure A.5 shows the velocity calculated using Eqs. (12.5) for the 25 zone problem. The velocities were averaged over a distance corresponding to the flame moving through one zone. It appears that the flame is pretty nearly in steady state at $200 \mu sec$. Figure A.6 shows a plot taken at this time. A comparison with Fig. A.4 supports the view that the later time represents the translation of the problem to the right about $0.1 cm$. Incidentally, during this time the problem has burned through two zones, even though the smallest

zone is less than $1.6 \times 10^{-3} \text{ cm}$ thick. This shows clearly how the grid is moving with the flame.

The chemistry calculation has first-order accuracy, and therefore we expect HCT to have approximately first-order accuracy as a whole. Thus the error should scale as $(\Delta x)^2$ or $(\Delta t)^2$. We ran a series of problems in which Δx and Δt decrease by a factor of approximately the square root of 2. Thus starting with 25 zones, we next ran with 35 zones and then 50 zones. Starting with an N^* of 5, we increased N^* to 7 and then 10 as the total number of zones increased. We also simultaneously reduced the relative change allowed each physical variable during a cycle from 0.1 to 0.071 to 0.05.

Figure A.7 shows a comparison of the velocity calculated from the three problems. If the errors scale as $(\Delta x)^2$ and $(\Delta t)^2$, the difference between curves {1} and {2} should be twice the difference between curves {2} and {3}. This is very nearly true. From this argument it follows that the true velocity should follow a curve the same distance below curve {3} as curve {3} is below curve {2}.

Figure A.8 shows the temperature profiles for all three problems at $400 \mu\text{sec}$. The left edge of the figure is the position of the original step in temperature, so the distance from the left edge is the distance the flame has travelled. All three profiles appear essentially the same, despite the fact that profile {3} is described by twice as many zones as profile {1}. The figure also shows a relative error in distance travelled consistent with the velocity differences of Fig. A.7.

We expect the problem running time to scale by Δt times Δr_{min} . The factor proportional to Δt is clear; the factor proportional to Δr follows since Δr is proportional to the amount of material in the zone. A small zone would then have the species changing more rapidly as the flame moved through.

The 25-zone problem took 38 seconds to run, the 35-zone problem 58 seconds. The 50-zone problem took 112 seconds, so the computer cost rises slightly more slowly than expected. This is partly because with smaller relative changes allowed per cycle the code requires fewer iterations to converge.

We also ran two more problems to separate the effects of decreasing the timestep from the effect of increasing the number of zones. Figure A.9 shows

the velocity curves from these additional runs plotted against the previous 25-zone problem (curve {1}) and the previous 35-zone problem (curve {4}). Run {2} is the same as run {1} except that the maximum relative change per timestep was decreased from 0.1 to 0.071. Run {3} differs from run {2} in that N^* was increased to 7 and the total number of zones increased to 27. Run {4} represents an increase in the total number of zones to 35, with all else the same as run {3}. It is clear that the greatest improvement in accuracy results from the increase in N^* .

The third example is designed to illustrate the behavior of the code when calculating a fairly complicated system with many species and reactions

The problem is a low-pressure (about 1/20 atmosphere) 9.5-percent (mole fraction) methane-oxygen mixture. The system is described by 15 species and 45 reactions on a 10-centimetre grid. There are 25 zones in the problem, with N^* equal to 8. Tables A.6 and A.7 show the species and reactions involved.

The problem starts with an initially uniform distribution. The left boundary is fixed, the right boundary open. The volume source of Eq. (9.1) is used. The parameters of Eq. (9.3) are a S_0^{vol} of $225 \text{ cal/cm}^3 \cdot \text{sec}$, x_0 equal 0 and Δx equal 0.04 cm . The source is on until t_{ig} equal 0.3 milliseconds . This is a slow enough source that no shocks are generated. Equation (9.7) implies that a total of $1.87 \times 10^{-3} \text{ cal/cm}^2$ are put into the problem.

The thermal conduction coefficient and species diffusion coefficients are again given by Eqs. (12.1) and (12.2).

Although the problem is initially uniform, we know that the source will concentrate the action at the left side of the problem. Furthermore, since the source is characterized by Eq. (9.1), we want to initialize the zoning with a grid that is capable of resolving the source as it dumps in energy. Thus the grid is initialized by the dynamic rezoning algorithm using an $f(x)$ for Eq. (6.2) in which dT/dx has been replaced by the source distribution function of Eq. (9.3). This gives the required resolution at the left side of the problem (see zone size plot of Fig. A.10).

Table A.8 gives the input deck for this problem. With the help of Appendix A one can see how the variables are specified for this problem.

One notices that some of the variables are set more than once. The NAMELIST input routines used by the code to read the input file always

stores the last value read. If variations are being run off a basic input deck, it is convenient to keep the basic deck unchanged and add the variations at the end. A simple check of the end of the input deck tells what changes have been made without the necessity of comparing the whole deck.

We point out how the species are specified from the data file CDAT. Variable LISTS specifies the species to be used according to their arrangement in the data file CDAT. The code then reads the appropriate data from the file and assigns a new species number according to the order in which the species were specified in the input deck. The numbers in LISTS need not be in any particular order. Thus LISTS specifies species 1 through 6 and 12 through 20 in CDAT. In the problem these species are renumbered 1 through 15, as can be seen in Table A.6. A similar scheme is used for the reaction list LISTR. Note that variables FRC, FRX, RRC and RRX reset some of the reaction constants, overriding the values obtained from CDAT. The index of the variable to be changed must correspond to the index the code uses, not the index in CDAT. A check with Table A.7 verifies that the correct parameters were used.

Figure A.10 shows the problem after cycle 1. Note that the zoning is concentrated at the left side of the problem. All species except O_2 and CH_4 have mole fractions around 10^{-6} .

Figure A.11 shows the left one-tenth of the grid at cycle 140. We show only the left tenth since very little is happening in the rest of the problem. At this stage we notice that enough energy has been added to heat the left boundary to over $2500K$. The heat added is pushing gas out the right side of the problem. It is also increasing the chemical reaction rates, producing significant concentrations of minor species.

Figure A.12 shows the same portion of the grid as Fig. A.11 just after the source has been turned off. From the velocity plot one sees that gas is being pushed away from the $0.1cm$ area, suggesting that energy production is largest there. The mole fraction plot, with its decreasing CH_4 concentration left of $0.1cm$, suggests that significant combustion has occurred there. These factors suggest a propagating flame. The temperatures higher than flame temperature at the left boundary are due to the source.

Figure A.13 shows the problem at nearly 30 *milliseconds*. Unlike Figs. A.11 and A.12, this shows the whole computational grid. The flame has propagated most of the way across and has settled into a nearly steady state.

Table A.9 give the overall edit of the problem at this time. The explanation of this edit was given for Table A.3. Table A.10 gives a printout of the species mole fractions at the time. This is a table of the information presented graphically in the lower right frame of Fig. A.13. Tables A.9 and A.10 together give a complete listing of the physical variables at this time.

EDOTR in Table A.9 shows that zone 13 is the zone with the largest rate of chemical heat release. Table A.11 gives an edit of the rates involved in changing the concentrations in this zone. This edit was explained for Tables A.4 and A.5, but in this case there is considerably more information to digest. Note that if every reaction was printed, rather than just the important ones, 90 reactions would be listed.

Table A.12 gives the relative change for the variables in the Lagrange part of the calculation. Since the timestep is based on the largest of these numbers, inspection of this edit gives an idea of whether the code is calculating efficiently or not. Table A.9 gives H_2O_2 in zone 13, with a relative change per cycle of 0.092, as the variable limiting the timestep. Table A.12 shows that all species except H_2O have relative changes of at least 0.01. Species H , O , OH , HCO , CH_3 , CH_4 , HO_2 , CH_2O , and CH_3O have relative changes of at least 0.05. This indicates that a calculational cycle represents a significant evolution of the physical state of the problem.

Table A.13 gives the relative change due to the remap. It allows an estimate of the contribution to change due to advection, although one must keep in mind that this is advection through a moving grid. In order to examine the advection through fixed points in the grid, we ran one more Eulerian cycle with the dynamic rezoner turned off. This causes the variables to be mapped back onto the original grid. Table A.14 shows the relative changes caused by advection alone for this additional cycle.

The numbers are not strictly comparable with data in Table A.13, since they refer to a different cycle. In addition, the change is calculated relative to the value of the variables at the end of the cycle. However, it is not necessary to correct for these effects to see a significant difference between the two tables.

If one looks at the zone with the largest Lagrange relative change for each species separately, one notes that the changes in Tables A.13 and A.14 are generally in the opposite direction. This is true for all species but H_2O , whose relative change is the smallest of all.

The dynamic rezoner is choosing a grid that tends to cancel the change produced by the Lagrange calculation. This is consistent with a picture of the grid moving with the flame front, so that the values of the physical variables in any zone are unchanged after each cycle.

In contrast, where the Lagrange change is largest for a species, the remap change shown in Table A.14 is in the same direction. The change due to advection adds in the same sense as the change produced by the Lagrange calculation.

In both cases, the changes produced by the remap are comparable to the changes due to the Lagrange calculation.

The fact that the changes in Tables A.12 and A.13 tend to cancel suggest an interesting mathematical question: Might it have been useful to include advection in the fundamental Eqs. (2.1), (2.9), (2.13) and (2.22), thus negating the necessity for a separate remap? This should allow a larger timestep, since the timestep is chosen to limit the total relative change in the species concentrations during the Lagrange calculation.

This was not done primarily for reasons of simplicity. The Lagrange equations are much simpler than the Eulerian equations with the several different types of remaps that we currently allow. To have a purely implicit Eulerian code with all remap possibilities, we would have to add coding to calculate the boundary terms given by Eqs. (5.3), (5.5), (5.9), (5.10), (5.11), (5.15), and (5.17), (as well as the auxilliary terms appearing in them) at the cycle end time in terms of the last value of the variables and the corrections to them.

In addition, one would have to make some guess as to the distance the grid should be moved if the dynamic rezoner was used. This may not prove difficult, however, as one could probably just move the grid to the configuration needed to describe the problem at the end of the previous cycle.

There is also the nontrivial question of stability of the proposed Eulerian set of fundamental equations. The Eulerian equations would not give the same change per timestep as is currently obtained by first doing the Lagrange cycle and then remapping, since the advection terms would be calculated using the same variables as the hydrodynamics, chemistry, and transport. The current coding uses an intermediate set of variables that are not equal

to those at the beginning of the cycle or those at the end of the cycle. They are those that exist after the Lagrange cycle, but before the remap. Conservation laws using these intermediate variables determine the remap. Since the Lagrange equations separately are stable and the remap by itself is stable, the combination is stable. The combined operations may introduce instabilities that are damped by the current intermediate step. This question needs to be investigated.

If the remap was done implicitly, the timestep limitation would probably become the requirement that the boundary of a zone not move through more than a given fraction of its neighboring zone in one cycle. This is a limitation we should expect since the difference scheme only connects nearest neighbors. For this example, we estimate the velocity of a zone boundary in the flame front to be approximately the flame velocity. This follows since the grid is tracking the flame. From Table A.9, this is at least 200 cm/sec in the problem frame of reference, since this is the velocity of the unburned gas. The flame is moving into the unburned gas, and thus must have a higher velocity than the unburned gas. That velocity would move the boundary between zones 13 and 14 through zone 14 in about $3.7 \times 10^{-4}\text{ seconds}$. If we allow a 10 percent transit in one cycle, this would give a timestep of $3.7 \times 10^{-5}\text{ seconds}$. This is nearly the current timestep. Thus the relative advantage of a pure Eulerian calculation is probably fairly problem dependent.

These are very interesting questions that we have not yet had the time to investigate. We leave them as questions to be answered in the future.

Dividing the timing information at the bottom of Table A.9 by the number of zones shows that an LU decomposition takes $4000\text{ }\mu\text{sec}/\text{zone} \cdot \text{call}$. An LU decomposition was called for on the average of twice every three cycles. The calls to SOL averaged $300\text{ }\mu\text{sec}/\text{zone} \cdot \text{call}$. Looking at Table A.3, this compares with $360\text{ }\mu\text{sec}/\text{zone} \cdot \text{call}$ and $80\text{ }\mu\text{sec}/\text{zone} \cdot \text{call}$ respectively for the ozone flame. The ozone flame with only 3 species has $M = 6$ zone variables, while the methane has $M = 18$. The increase by a factor of 11 in the cost of DEC is less than the factor of 27 one would expect if the LU decomposition scaled as M^3 , but is more than a factor of M^2 . Apparently one has not reached the asymptotic limit. The SOL calls scale by a factor of 3.8, compared to the factor of 9 expected if they scaled by M^2 .

One notes that calls to DIFPED are about an order of magnitude more expensive in the methane flame than in the ozone flame. Since the number of

reactions involved increased from 3 to 45, this is consistent with regarding DIFPED as scaling linearly with the number of reactions. Thus while in the smaller problem the LU decompositions are less expensive than the calculation of the equations, in the methane flame the matrix routines are more expensive. It is clear that as one calculates problems with greater numbers of species, the matrix routines will consume a larger fraction of the total computer time.

We also note that DIFPED is 4 times as expensive as DIFFUN, i.e., the left side of Eq. (4.5) costs 4 times as much to evaluate as the right side. In addition, the LU decomposition is over 13 times as expensive as a call to SOL to solve the equations by back substitution once DEC has been called. In the actual running of the problem, the time spent in DIFPED plus DEC and SOL is roughly comparable to the time spent in DIFFUN plus SOL. Thus the use of Eq. (4.8) when possible has considerably reduced the running time of the problem. This is not completely clear from this argument alone, since evaluation of a new matrix T^n each iteration gives faster convergence, but an independent check indicates the current method of using Eq. (4.8) when possible is probably saving at least a factor of two in running time. There may be additional ways to minimize running time that we have not explored. For example, one might want to try different convergence criteria, or different ways of determining when a new matrix T^n should be calculated.

From the total CPU time spent we see that the problem took almost eight minutes to run to the configuration of Fig. A.13. This is a cost of about $12.5 \text{ milliseconds}/\text{zone} \cdot \text{cycle}$. The corresponding figure for the ozone flame is $1.5 \text{ milliseconds}/\text{zone} \cdot \text{cycle}$.

This problem was actually run in such a way as to illustrate two distinct problems—ignition and steady state propagation. If one is interested mainly in ignition, the calculation can be terminated at cycle 300, as seen in Fig. A.12. The calculation of the first 300 cycles costs about 1.8 minutes.

If one is interested in seeing the effects of various changes in physical parameters on steady state calculations, one could run the first calculation as a baseline problem. Other runs could start from the dump of this problem at cycle 1500, modify the parameters, and run a few hundred cycles until the transients settle out. This would be a considerably less expensive procedure than running the whole problem over each time from ignition. The whole HCT code system is built to work easily in this mode.

Relation To Other Methods

There exists a literature on computer methods to solve the time-dependent hydrodynamics, kinetics, and transport equations (in one dimension) that are described in this report. In this section we discuss the relation between HCT and some of these methods.

Spalding first proposed using the time-dependent equations solved here to predict flame speeds.[11] Aside from assuming pressure equilibrium, the basic equations are similar. We just point out that including pressure variations allows one to solve a wider class of problems. As shown earlier, we have written our difference equations in such a way that uniform pressure problems are also calculated correctly.

Spalding proposed to calculate the flame speed by letting an initial configuration settle to a steady state by solving the time-dependent equation. We illustrated this method in calculating the flame speed for our ozone flame. Several workers have used this basic idea in several investigations.[12],[13],[14] However, the methods used for differencing the equations vary, as well as the method of solution of the resulting difference equations.

Our hydro equations are based on the scheme used in KRAKEN.[15] The most distinctive feature we have borrowed is the consideration of an Eulerian calculation as a two-step process—a Lagrange calculation followed by a remap of the physical variables. This method makes dynamic rezoning particularly simple. Our use of a Newton's iteration method to solve implicitly differenced hydrodynamics equation is similar to a method used by Lindemuth.[16]

The general hydrodynamic and transport equations are written in an explicitly conservative form. Thus the species concentrations are automatically conserved, the hydrodynamic work done on the fluid goes into the internal energy of the species, and momentum is conserved in the interior of the mesh. Thus even if the mesh has varying degrees of resolution in different parts of the problem, these important characteristics of the problem will be preserved.

We note that species are conserved to the order of machine roundoff by the Newton's iteration scheme we use. This is true even if one is iterating using the old matrix T^m , as in Eq. (4.8), rather than evaluating a new T^n , as in Eq. (4.5). This follows from the fact that the linearized equations we solve to machine roundoff conserve concentrations. Using Eq. (4.8) is equivalent to using Eq. (4.5) with incorrect derivatives in the expansion of the implicit terms about the latest variable values. This expansion still has the property of conserving species. This may not be true in other iteration schemes.

Our equations are differenced directly on the space coordinate, even though the equations are written in the Lagrange reference frame. Bledjian[12] worked explicitly in the Lagrange coordinate; Spalding, Stephenson and Taylor[13] and Smoot, Hecker and Williams[14] make an additional variable transformation to handle boundaries at infinity.

Our equations are all differenced implicitly in terms of the values at the end of the calculational cycle. It is well understood that this is necessary for an efficient computer algorithm if the difference equations are sufficiently stiff. The usual reason for reluctance in using such equations is that solving the resulting implicit relationships can be very expensive, since they involve inverting large matrices. We have shown that this expense is affordable for the problems considered. The fact that the time spent calculating the physics in DIFFUN is comparable to the time spent in the matrix inversion routines illustrates this. This will not be true for sufficiently large problems, of course. They may have to await even faster computers than the CDC 7600 to be practical.

There are ways to have the advantages of implicitly differenced equations without having a system of equations such as those in HCT. The simplest is operator-splitting, where the individual operators are implicit but are explicitly coupled.[17] In one such scheme, the hydro and transport equations are differenced implicitly but without the chemistry terms.[18] The implicit hydro equations are solved and a separate subroutine called to update the chemistry. The hydro scheme uses a generalized ICE method,[19] the chemistry uses a one-step backward Euler scheme.[20] Thus symbolically, we have

$$\frac{d\vec{X}(t)}{dt} = A_{hydro}(\vec{X}(t)) + B_{chem}(\vec{X}(t)) \quad (13.1)$$

In an operator-split algorithm, this equation is solved in two steps, each of which is implicit:

$$\vec{X}^{n+\frac{1}{2}} - \vec{X}^n = \Delta t A_{hydro}(\vec{X}^{n+\frac{1}{2}}) \quad (13.2)$$

is followed by

$$\vec{X}^{n+1} - \vec{X}^{n+\frac{1}{2}} = \Delta t B_{chem}(\vec{X}^{n+1}). \quad (13.3)$$

Adding these two equations gives

$$\vec{X}^{n+1} - \vec{X}^n = \Delta t \left[A_{hydro}(\vec{X}^{n+\frac{1}{2}}) + B_{chem}(\vec{X}^{n+1}) \right]. \quad (13.4)$$

It is clear that each individual equation is implicit, and thus the stiffness of each separate operator is taken into account. If $\vec{X}^{n+1/2}$ is intermediate between \vec{X}^n and \vec{X}^{n+1} , the method may be as accurate as a fully implicit scheme that solves

$$\vec{X}^{n+1} - \vec{X}^n = \Delta t \left[A_{hydro}(\vec{X}^{n+1}) + B_{chem}(\vec{X}^{n+1}) \right] \quad (13.5)$$

in one step. Furthermore, it is clear what one must do to be sure that the two-step procedure of Eqs. (13.2) and (13.3) be as accurate as solving Eq. (13.5). We just have to make sure that the changes in going from \vec{X}^n to $\vec{X}^{n+1/2}$ and from $\vec{X}^{n+1/2}$ to \vec{X}^{n+1} are comparable to the changes allowed in going directly from \vec{X}^n to \vec{X}^{n+1} .

The choice of the operators to be split is usually made on the basis of which combinations of operators are easiest to solve. The most serious problems arise if the mathematics dictates a natural choice of A and B, but this choice gives large changes that tend to cancel. In this case the coupling between the two operators is stiff. The timestep must be limited to give a small change in either of the operators, rather than the sum. If one is trying to calculate a quantity such as flame velocity that depends on accurately calculating the interplay between the two operators, one can get into timestep troubles.

Even with these restrictions, operator-split codes have been used to calculate solutions to a variety of combustion problems. We have chosen to solve the fully implicit system for the added generality of being able to efficiently solve problems where operator splitting is known to have difficulty. This is in keeping with our general philosophy of using as “robust” an integrator as possible. Tables A.4, A.5, and A.11, show examples where

the chemistry and hydrodynamics are closely coupled. For the examples we have tested, our choice has not resulted in significantly larger computational costs.

Another approach is to solve the implicitly differenced Eq. (13.5) with an iterative matrix inverter. This gives the advantage of the fully implicit difference equations. An iterative algorithm may have advantages in computer storage requirements or computer costs considerations. In the case of two-dimensional problems,[21] one seems forced to such a method since the LU decomposition of the matrix analogous to T^n in Eq. (4.5) requires considerably more storage than the original matrix. It also requires proportionally more calculations. We merely point out that there is a noniterative method for the one-dimensional equations. Our examples show it to be practical for our applications. We know of no iterative method that shows promise in this situation.

A more general approach to solving Eqs. (3.3), (3.8) and (3.16) is the method of lines.[22] Included in this general idea is the idea of reducing the partial differential equations to a set of ordinary differential equations (ODE's) by a particular discretization, and then solving the resulting set of ODE's using a standard ODE solver such as GEAR.[23]

One of the important phases of this process is the reduction to a set of ODE's. Conceptually, this is the reverse of the way we derive our equations, i.e., we first wrote our description of the system in finite form and then showed that it reduces to the appropriate partial differential equations.

It is apparent that one introduces errors in the process of spatial discretization. In general, the errors introduced here will set an upper limit on the accuracy of the result. This would suggest that there is little to be gained in integrating the resulting ODE's to an accuracy greater than that inherent in the discretization.

It has been proposed that one can fit the variables to a well-defined spatial order of accuracy by finite element methods. Margolis has used B-spline fits giving fourth-order accuracy in distance, combined with the ODE-solver package GEAR to calculate a propagating ozone flame.[24] The whole procedure is done using the software package PDECOL of Madsen and Sinovec.[25]

This procedure does not ensure that the conservation laws are satisfied exactly, as the finite difference equations used in HCT do. However, fourth-order accuracy in Δx should be sufficiently accurate if the grid is fine enough to follow the variations of all the species.

A higher order description will in general require more grid points than a lower order description to insure a well-behaved solution. For example, a first-order fit between two grid points of a function that is really a decreasing exponential will at least be positive definite. A higher order polynomial fit through several points may not. HCT uses lower order approximations in order to give well-behaved solutions with as few grid points as possible. If the number of grid points required is significantly less than the number required for the higher order schemes, the calculations will be correspondingly less expensive. The key question is whether the lower order equations converge rapidly enough to the true solution to be useful.

Table A.10 for the methane flame shows several concentrations that vary almost an order of magnitude from zone to zone. Note, for example, how quickly the CH_3 and CH_4 concentrations fall off behind the flame front. Such variations can not be fit well with so few grid points by any interpolation higher than first order, and yet useful answers can be obtained from such calculations. This can be seen from the ozone flame calculation, where the 25-zone problem gave a flame velocity to within about 10 percent, even with such a crude representation of the species concentrations.

A higher order interpolation also has the disadvantage of connecting more than 3 grid points in the difference equations. Such a circumstance results in a matrix equation analogous to Eq. (4.5) that is larger than block tridiagonal, considerably increasing both storage requirements and the cost of solving the equations. We prefer a system of relatively low formal accuracy where one increases the number of zones and reduces the timestep for increased accuracy. One then experiments with these parameters until the desired accuracy is obtained.

HCT has another feature that makes the method of lines more difficult to apply in this case. This feature is the variable grid algorithm that chooses the grid on the bases of the temperature distribution. It is undoubtedly possible to use the method of lines with a variable grid, but we have not had time to investigate the properties of such a system.

HCT might benefit from the more sophisticated error-control procedures of general software packages such as GEAR. This may allow larger timesteps than our method uses. It may also need fewer iterations to converge to the solution of the implicitly differenced equations.

One might think that since we also calculate the Jacobian of the equations of motion, one could use HCT's routines in the GEAR package, using the simpler discretization that HCT provides. In fact, one familiar with the GEAR package will notice a similarity in the notation used in this report. To do this correctly, one should difference the transport coefficients and any other explicit terms at the cycle end time. This is not done now.

In summary, HCT is a conventional finite difference code. Accuracy is shown by the heuristic, but straightforward, method of trying calculations with different numbers of zones and different timesteps. The variable grid aids in getting resolution where needed while still keeping the total number of zones low. We use fully implicit difference equations that give as "robust" an integrator as possible.

Acknowledgements

It is a pleasure to acknowledge many useful discussions of combustion phenomena with Jim Routh, John Creighton, Len Haselman and Charles Westbrook.

The hydrodynamic scheme used here derived several useful features from discussions with Jim Leblanc. It is impossible to acknowledge all the computational hydrodynamics literature containing ideas we have used here to solve the hydrodynamics equations. We have merely cited those papers that were most directly useful to us.

References

- [1] P. J. Roache, Computational Fluid Dynamics (Hermosa Publishers, Albuquerque, NM, 1976).
- [2] JANAF Thermochemical Tables, Dow Chemical Company, Midland, Michigan (1974).
- [3] F. A. Williams, Combustion Theory (Addison-Wesley Publishing Company, Inc., Reading, MA, 1965).
- [4] A. C. Hindmarsh,
Solution of Block-Tridiagonal Systems of Linear Algebraic Equations, Lawrence Livermore Laboratory, Livermore, CA, UCID-30150 (1977).
- [5] A. C. Hindmarsh, L. J. Sloan, K. W. Fong and G. H. Rodrigue,
DEC/SOL: Solution of Dense Systems of Linear Algebraic Equations, Lawrence Livermore Laboratory, Livermore, CA, UCID-30137 (1976).
- [6] R. Courant and D. Hilbert, Methods of Mathematical Physics Vol. 1 (Interscience Publishers, New York, NY, 1953).
- [7] C. K. Westbrook and L. L. Chase,
Chemical Kinetics and Thermochemical Data for Combustion Applications, Lawrence Livermore Laboratory, Livermore, CA, UCID-17833, Rev. 4 (1988).
- [8] F. H. McMahon, L. J. Sloan and G. A. Long,
STACKLIB: A Vector Function Library of Optimum Stack-Loops for the CDC 7600, Lawrence Livermore Laboratory, Livermore, CA, UCID-30083 (1972).
- [9] J. T. Martin, R. G. Zwakenberg and S. V. Solbeck,
LRLTRAN Language Used With the CHAT and STAR Compilers, Lawrence Livermore Laboratory, Livermore, CA, LTSS-207, 4th ed (1974).
- [10] J. H. Rowe, DISPLAY, Lawrence Livermore Laboratory, Livermore, CA, UCID-30081 (1973).
- [11] D. B. Spalding, "The Theory of Flame Phenomena with a Chain Reaction," Phil. Trans. Roy. Soc. London, A249, 1 (1956).

- [12] L. Bledjian, "Computation of Time-Dependent Laminar Flame Structure," Combustion and Flame, **20**, 5 (1973).
- [13] D. B. Spalding, P. L. Stephenson and R. G. Taylor, "A Calculation Procedure for the Prediction of Laminar Flame Speeds," Combustion and Flame, **17**, 55 (1971).
- [14] L. D. Smoot, W. C. Hecker and G. A. Williams, "Prediction of Propagating Methane-Air Flames," Combustion and Flame, **26**, 323 (1976).
- [15] R. B. Debar, Fundamentals of the KRAKEN Code, Lawrence Livermore Laboratory, Livermore, CA, UCIR-760 (1974).
- [16] I. R. Lindemuth, The Alternating-Direction Implicit Numerical Solution of Time-Dependent, Two-Dimensional, Two-Fluid Magnetohydrodynamics Equations, Lawrence Livermore Laboratory, Livermore, CA, UCRL-51103 (1971).
- [17] R. Richtmyer and K. Morton, Difference Methods for Initial-Value Problems (Interscience Publishers, New York, NY, 1967), 2nd ed.
- [18] C. K. Westbrook and L. L. Chase, A One-Dimensional Combustion Model, Lawrence Livermore Laboratory, Livermore, CA, UCRL-52297 (1977).
- [19] F. H. Harlow and A. A. Amsden, "A Numerical Fluid Dynamics Calculation Method for All Flow Speeds," J. Comp. Physics, **8**, 197 (1971).
- [20] C. M. Lund, Lawrence Livermore Laboratory, unpublished notes.
- [21] J. S. Chang, A. C. Hindmarsh and N. K. Madsen, Simulation of Chemical Kinetics Transport in the Stratosphere, Lawrence Livermore Laboratory, Livermore, CA, UCRL-7483 (1973).
- [22] N. K. Madsen and R. F. Sincovec, The Numerical Method of Lines for the Solution of Nonlinear Partial Differential Equations, Lawrence Livermore Laboratory, Livermore, CA, UCRL-75142 (1973).
- [23] A. C. Hindmarsh, GEAR: Ordinary Differential Equation System Solver, Lawrence Livermore Laboratory, Livermore, CA, UCID-30001 Rev. 3 (1974).

-
- [24] S. B. Margolis,
Time Dependent Solution of a Premixed Laminar Flame, Sandia
Laboratory, Livermore, CA, SAND77-8506 (1977).
- [25] N. K. Madsen and R. F. Sincovec, PDECOL:
General Collocation Software for Partial Differential Equations,
Lawrence Livermore Laboratory, Livermore, CA, UCRL-78263 Rev. 1
(1977).

Appendix A

Sample Output

Figure A.1: Test problem involving only hydrodynamics and thermal conduction after initial cycle.

Figure A.2: Test problem involving only hydrodynamics and thermal conduction after 50 cycles.

```

lists = 3 4 28

cpa1(1) = 5.365 cpa2(1) = -9.186e-1 cpa3(1) = 8.955e-1
cpa4(1) = -4.39e-1 cpa5(1) = 1.06e-1 cpa6(1) = -9.8667e-3

listr = 14 76 77

frc(1) = 2.5e19 frp(1) = -1. frx(1) = 1.19e5
rrc(1) = 1.3e17 rrp(1) = -.75 rrx(1) = 0.
rrc(2) = 0. rrp(2) = 0. rrx(2) = 0.
frc(3) = 9.94e14 frp(3) = 0. frx(3) = 2.2652e4
rrc(3) = 1.67e13 rrp(3) = 0. rrx(3) = -2.0860e3

c0 = 1.e-2 .99 2.e-5 1250. .004 0.
c1 = 1.e-6 .75 .25 300. .004 0.
xdisc = .02
xsprd = .001

pr0 = 1.e6
prbr = 1.e6
nzones = 25
nzxcass = 5
xmax = .3

dzero = 4.58e-7
dczero = 4.16e-7

iprint = 50 0 10 50 50 50
maxdump = 200
lste4 = 1:25
itimer = 1

eta2 = .15
delymn = 20.
euler = 1.
euler2 = -1.
rzflag = 1.
fnertia = .001
itermx = 20
rdelcmx = .1
dt = 1.e-9
dtmax = 1.
timemx = .0004
dtmin = 1.e-20
mxcycle = 1990

$
$

```

Table A.1: Input deck used to generate ozone test problem.

i	specie		atwt	cpa1	cpa2	cpa3	cpa4	cpa5	cpa6	h0
1	o	[3]	16.	5.3650e+00	-9.1860e-01	8.9550e-01	-4.3900e-01	1.0600e-01	-9.8667e-03	5.9550e+01
2	o2	[4]	32.	5.1260e+00	6.9420e+00	-5.8620e+00	2.6840e+00	-6.1000e-01	5.4670e-02	0.
3	o3	[28]	48.	5.5610e+00	1.8090e+01	-1.6810e+01	8.0620e+00	-1.9340e+00	1.8400e-01	3.4100e+01

Table A.2: Data associated with the equations of state used in the ozone test problem.

k	reaction				fre	frp	frx	rrc	rrp	rrx
1	o2	=	o+o	[14]	2.500e+19	-1.00	1.190e+05	1.300e+17	-0.75	0.
2	o+o3	=	o2+o2	[76]	3.370e+13	0.00	5.700e+03	0.	0.00	0.
3	o3	=	o+o2	[77]	9.940e+14	0.00	2.265e+04	1.670e+13	0.00	-2.086e+03

k	reaction				irf			irs		
1	o2	=	o+o	[14]	4	2	1	0	0	0
2	o+o3	=	o2+o2	[76]	2	1	3	2	0	0
3	o3	=	o+o2	[77]	3	3	1	2	0	0

Table A.3: Data associated with the reaction set used in the ozone test problem.

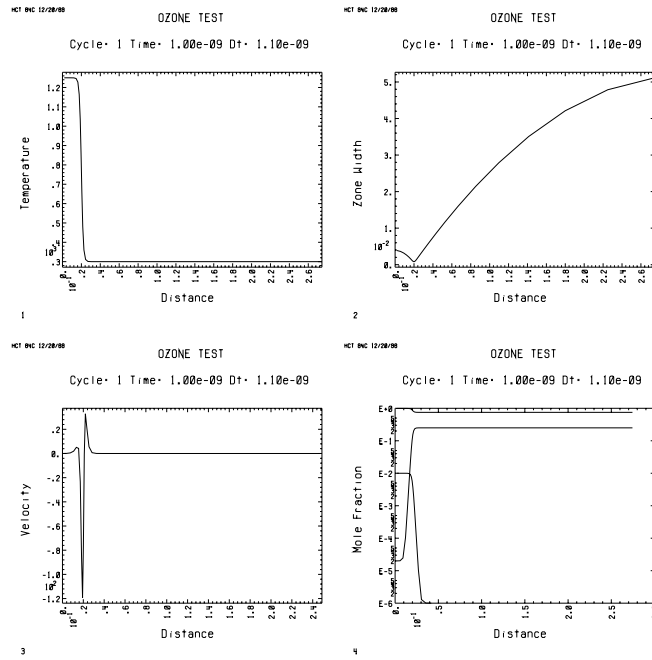


Figure A.3: Ozone test problem after initial cycle.

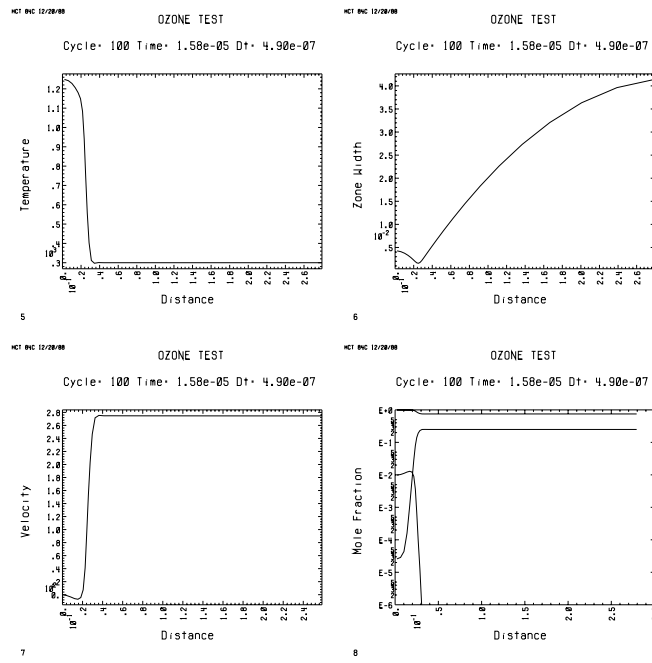


Figure A.4: Ozone test problem after 100 cycles.

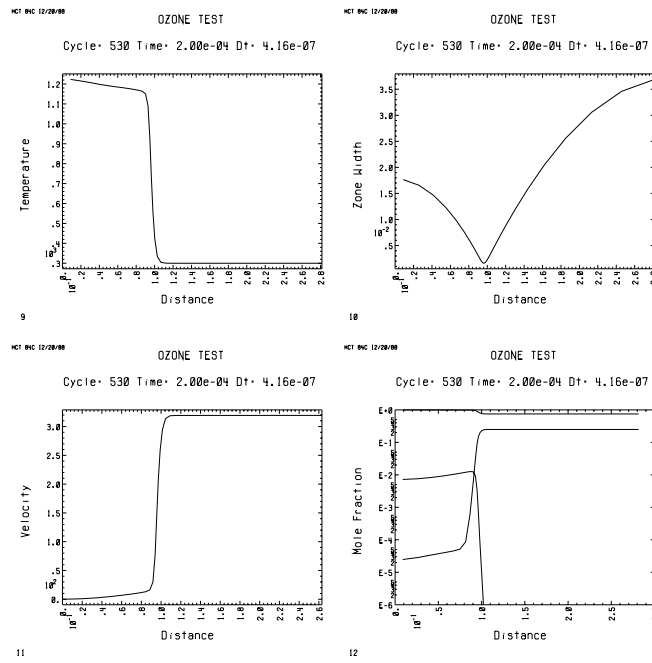


Figure A.5: Ozone test problem after flame has propagated past the middle of the calculational grid.

```
cycle=1000 avgiter=3.41e+00 ircytot=0 irzftot=0 njcibtot=271 time=3.8866e-04 dt=3.96e-07
```

```
species limiting dt- 3 zone- 11
slowest converging species- 1 zone- 19
```

jd	r	dr	ctot	rho	t	u	press	edotr	edotc	edotd	p*dv/dx	vsound
1	0.	3.25e-02	9.78e-06	3.12e-04	1230.01	0.	1.00e+06	2.19e+08	-6.02e+07	3.00e+05	-3.39e+07	6.47e+04
2	3.2503e-02	3.04e-02	9.86e-06	3.15e-04	1219.23	1.10e+00	1.00e+06	2.71e+08	2.76e+06	3.50e+05	-6.00e+07	6.44e+04
3	6.2898e-02	2.67e-02	9.94e-06	3.17e-04	1209.81	2.92e+00	1.00e+06	3.69e+08	1.58e+07	3.81e+05	-8.46e+07	6.42e+04
4	8.9602e-02	2.22e-02	9.99e-06	3.19e-04	1203.53	5.18e+00	1.00e+06	5.07e+08	-2.18e+07	4.29e+05	-1.06e+08	6.40e+04
5	1.1180e-01	1.76e-02	1.00e-05	3.20e-04	1196.70	7.54e+00	1.00e+06	6.83e+08	-4.75e+07	6.36e+05	-1.39e+08	6.39e+04
6	1.2940e-01	1.35e-02	1.01e-05	3.22e-04	1189.03	9.99e+00	1.00e+06	8.94e+08	-6.09e+07	9.90e+05	-1.83e+08	6.37e+04
7	1.4286e-01	1.00e-02	1.02e-05	3.24e-04	1181.49	1.25e+01	1.00e+06	1.13e+09	-7.85e+07	1.37e+06	-2.32e+08	6.35e+04
8	1.5286e-01	7.27e-03	1.02e-05	3.26e-04	1174.70	1.48e+01	1.00e+06	1.41e+09	-1.26e+08	-1.31e+05	-2.84e+08	6.34e+04
9	1.6012e-01	5.20e-03	1.03e-05	3.27e-04	1168.75	1.68e+01	1.00e+06	2.66e+09	-7.27e+08	-4.99e+07	-3.94e+08	6.32e+04
10	1.6533e-01	3.68e-03	1.04e-05	3.29e-04	1161.46	1.89e+01	1.00e+06	1.84e+10	-1.02e+10	-7.03e+08	-1.59e+09	6.30e+04
11	1.6901e-01	2.60e-03	1.06e-05	3.39e-04	1134.68	2.47e+01	1.00e+06	1.23e+11	-7.62e+10	-4.22e+09	-1.04e+10	6.20e+04
12	1.7161e-01	1.90e-03	1.17e-05	3.82e-04	1032.03	5.17e+01	1.00e+06	.94e+11	-1.65e+11	-5.56e+09	-3.30e+10	5.84e+04
13	1.7351e-01	1.60e-03	1.43e-05	4.87e-04	837.93	1.14e+02	1.00e+06	1.41e+11	2.26e+10	2.00e+09	-4.36e+10	5.16e+04
14	1.7511e-01	1.64e-03	1.85e-05	6.46e-04	648.30	1.84e+02	1.00e+06	2.07e+10	1.15e+11	2.13e+09	-3.48e+10	4.49e+04
15	1.7675e-01	1.98e-03	2.46e-05	8.71e-04	489.42	2.41e+02	1.00e+06	1.47e+09	8.64e+10	1.08e+09	-2.25e+10	3.89e+04
16	1.7873e-01	2.66e-03	3.24e-05	1.16e-03	370.80	2.86e+02	1.00e+06	5.46e+07	4.24e+10	3.74e+08	-1.11e+10	3.40e+04
17	1.8138e-01	3.71e-03	3.85e-05	1.38e-03	312.56	3.15e+02	1.00e+06	1.63e+06	1.16e+10	6.28e+07	-3.08e+09	3.13e+04
18	1.8510e-01	5.19e-03	4.00e-05	1.44e-03	300.88	3.27e+02	1.00e+06	2.00e+04	1.21e+09	3.71e+06	-3.24e+08	3.08e+04
19	1.9029e-01	7.15e-03	4.01e-05	1.44e-03	300.04	3.28e+02	1.00e+06	1.11e+02	4.56e+07	8.24e+04	-1.22e+07	3.07e+04
20	1.9743e-01	9.64e-03	4.01e-05	1.44e-03	300.01	3.28e+02	1.00e+06	1.54e+00	1.08e+06	1.47e+03	-2.89e+05	3.07e+04
21	2.0707e-01	1.26e-02	4.01e-05	1.44e-03	300.00	3.28e+02	1.00e+06	1.24e+00	4.27e+04	7.71e+01	-1.17e+04	3.07e+04
22	2.1970e-01	1.59e-02	4.01e-05	1.44e-03	300.00	3.28e+02	1.00e+06	1.24e+00	3.75e+03	9.20e+00	-1.26e+03	3.07e+04
23	2.3564e-01	1.92e-02	4.01e-05	1.44e-03	300.00	3.28e+02	1.00e+06	1.24e+00	5.48e+02	1.38e+00	-4.35e+02	3.07e+04
24	2.5484e-01	2.19e-02	4.01e-05	1.44e-03	300.00	3.28e+02	1.00e+06	1.24e+00	3.84e+01	2.26e-01	-3.42e+02	3.07e+04
25	2.7672e-01	2.34e-02	4.01e-05	1.44e-03	300.00	3.28e+02	1.00e+06	1.24e+00	4.64e+01	5.80e-02	-2.09e+02	3.07e+04
total energy production rate							1.30e+09	-3.09e-06	-1.42e+07	-3.28e+08		

```
total mass= 2.330746e-04 total internal energy= 2.015386e+06
```

```
time spent (sec)-cpu= 2.526e+01 i-o= 4.742e+00 sys= 2.218e+00
```

subroutine	sec/call	sec/cycle
diffun	1.89e-03	5.93e-03
difped	4.40e-03	1.19e-03
solbtl	9.89e-04	3.37e-03
decbtl	5.68e-03	1.54e-03
rezone	9.10e-04	9.10e-04
ckcnvg	2.62e-04	8.94e-04
remap	1.26e-03	1.26e-03

Table A.4: General Lagrange edit of ozone test problem at cycle 1000.

zone 12	reaction	rate	o	o2	o3
	(concentration)		7.75e-08	1.09e-05	6.45e-07
	total rate		2.31e-01	8.55e-01	3.57e-01
	net rate		9.82e-03	-1.95e-01	-1.36e-01
	diffusion		-1.31e-02	-1.91e-01	3.08e-01
	hydrodynamics	-3.303e+04	-1.11e-02	-4.22e-01	-5.98e-02
	chemistry		6.66e-02	3.85e-01	-6.29e-01
3	o3=o+o2	1.205e-01	5.21e-01	1.41e-01	-3.38e-01
2	o+o3=o2+o2	1.046e-01	-4.52e-01	2.45e-01	-2.93e-01
zone 12	reaction	rate	temp		
	(concentration)		1032.03		
	total rate		2.29e+08		
	net rate		2.79e+07		
	conduction		-2.23e-01		
	diffusion		-7.52e-03		
	hydrodynamics	-3.303e+04	-4.47e-02		
	chemistry		3.97e-01		
3	o3=o+o2	1.205e-01	-1.64e-01		
2	o+o3=o2+o2	1.046e-01	5.60e-01		

Table A.5: Lagrange edit of rates affecting the physical variables in zone 12 for ozone test problem at cycle 1000.

zone 4	reaction	rate	o	o2	o3
	(concentration)		7.39e-08	9.92e-06	2.95e-10
	total rate		6.72e-04	1.75e-03	5.86e-04
	net rate		-2.03e-04	-9.54e-04	-9.61e-07
	diffusion		1.30e-02	-2.54e-03	7.66e-05
	hydrodynamics	-1.062e+02	-1.17e-02	-6.03e-01	-5.35e-05
	chemistry		-3.03e-01	5.92e-02	-1.66e-03
-3	o+o2=O3	2.923e-04	-4.35e-01	-1.67e-01	4.99e-01
3	O3=O+O2	2.256e-04	3.36e-01	1.29e-01	-3.85e-01
2	O+O3=O2+O2	6.772e-05	-1.01e-01	7.76e-02	-1.16e-01
-1	O+O=O2	3.467e-05	-1.03e-01	1.99e-02	0.
zone 4	reaction	rate	temp		
	(concentration)		1203.53		
	total rate		3.98e+05		
	net rate		1.40e+05		
	conduction		-2.01e-02		
	diffusion		3.96e-04		
	hydrodynamics	-1.062e+02	-9.81e-02		
	chemistry		4.68e-01		
-3	o+o2=O3	2.923e-04	2.68e-01		
3	O3=O+O2	2.256e-04	-2.06e-01		
2	O+O3=O2+O2	6.772e-05	2.48e-01		
-1	O+O=O2	3.467e-05	1.59e-01		

Table A.6: Lagrange edit of rates affecting the physical variables in zone 4 for ozone test problem at cycle 1000.

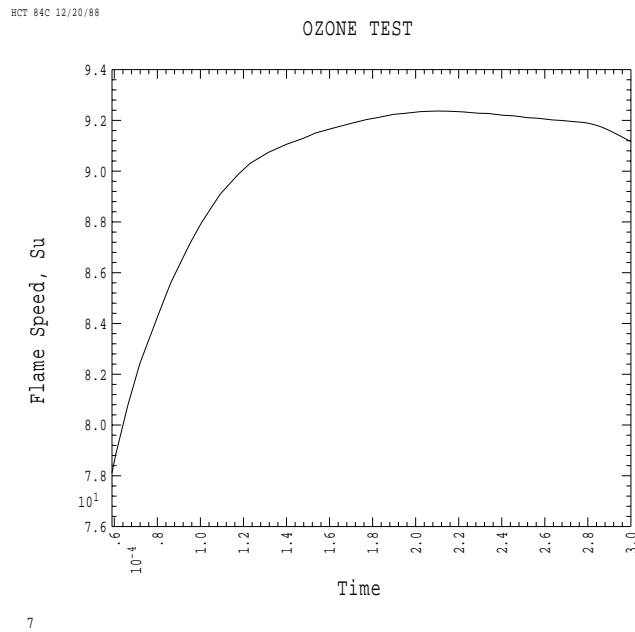


Figure A.6: Velocity of flame relative to the unburned gas in ozone test problem.

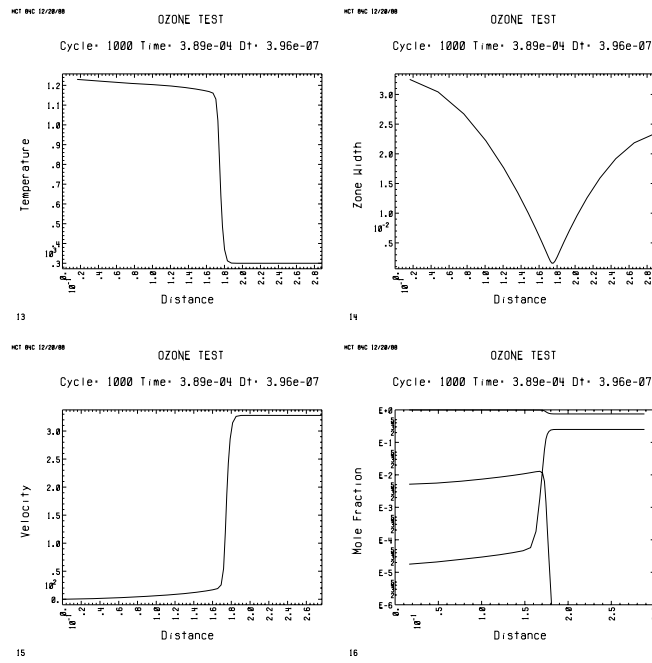


Figure A.7: Ozone test problem just after steady-state propagating flame has been established.

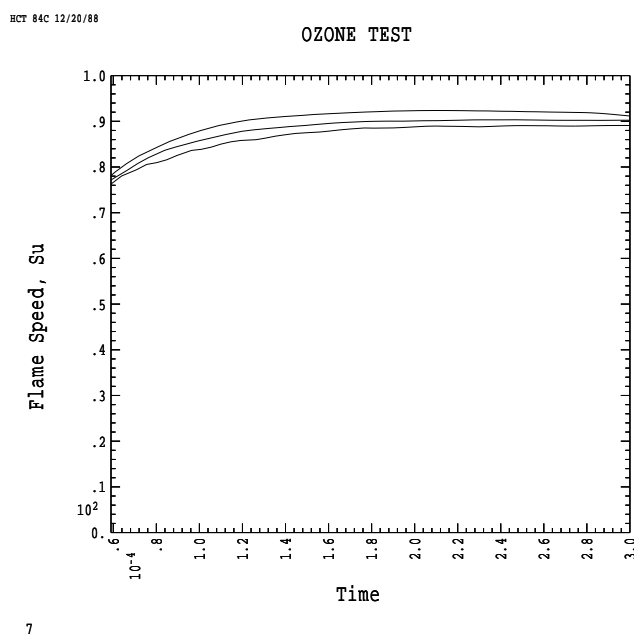


Figure A.8: Comparison of calculated flame velocities as a function of time for 25-zone ozone test problem (curve {1}) with velocities from calculations of the same problem run with 35 zones (curve {2}) and 50 zones (curve {3}).

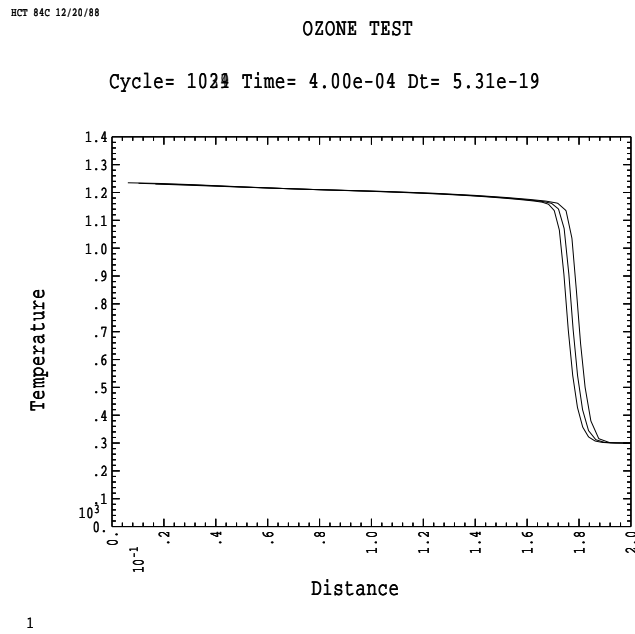


Figure A.9: Comparison of calculated temperature profile at $400\mu sec$ for 25-zone ozone test problem (curve {1}) with profiles from calculations of the same problem using 35 zones (curve {2}) and 50 zones (curve {3}).

Figure A.10: Comparison of calculated flame velocities as a function of time for the 25-zone ozone test problem (curve {1}) with calculations of the same problem with varying running parameters. Curve {2} represents a calculation with reduced timesteps, curve {3} represents a calculation with more zones in the reaction region, and curve {4} represents a calculation with more total zones.

cdat version		149	made		11/01/88					
i	specie		atwt	cpa1	cpa2	cpa3	cpa4	cpa5	cpa6	h0
1	h	[1]	1.	4.9680e+00	0.	0.	0.	0.	0.	5.2100e+01
2	h2	[2]	2.	7.7390e+00	-3.1923e+00	4.3373e+00	-2.1040e+00	4.8267e-01	-4.3733e-02	0.
3	o	[3]	16.	5.3650e+00	-9.1860e-01	8.9550e-01	-4.3900e-01	1.0600e-01	-9.8667e-03	5.9553e+01
4	o2	[4]	32.	5.1260e+00	6.9420e+00	-5.8620e+00	2.6840e+00	-6.1000e-01	5.4670e-02	0.
5	oh	[5]	17.	8.0760e+00	-4.4880e+00	6.4050e+00	-3.4300e+00	8.5070e-01	-8.1330e-02	9.4920e+00
6	h2o	[6]	18.	8.0490e+00	-1.6200e+00	6.4140e+00	-3.8950e+00	9.9930e-01	-9.6270e-02	-5.7800e+01
7	co	[12]	28.	6.0700e+00	2.1550e+00	1.9370e-01	-7.2900e-01	2.7330e-01	-3.2270e-02	-2.6420e+01
8	hco	[13]	29.	5.0870e+00	1.0950e+01	-5.9930e+00	1.6200e+00	-1.8870e-01	4.8000e-03	1.0400e+01
9	co2	[14]	44.	5.2750e+00	1.5610e+01	-1.1870e+01	4.9420e+00	-1.0740e+00	9.5200e-02	-9.4050e+01
10	ch3	[15]	15.	7.0280e+00	7.1880e+00	2.1020e+00	-3.1530e+00	1.0390e+00	-1.1440e-01	3.4820e+01
11	ch4	[16]	16.	1.8010e+00	2.1890e+01	-6.5910e+00	-4.8770e-01	6.3530e-01	-9.1470e-02	-1.7890e+01
12	ho2	[17]	33.	5.8310e+00	9.8920e+00	-6.2280e+00	2.2960e+00	-4.6500e-01	3.9700e-02	5.0000e+00
13	h2o2	[18]	34.	9.3600e+00	6.1930e+00	1.9430e+00	-3.7270e+00	1.4300e+00	-1.7600e-01	-3.2530e+01
14	ch2o	[19]	30.	2.3830e+00	2.1200e+01	-1.1630e+01	3.2870e+00	-4.4700e-01	2.1100e-02	-2.7700e+01
15	ch3o	[20]	31.	4.2400e-01	3.1660e+01	-1.8800e+01	6.1520e+00	-1.0660e+00	7.6500e-02	3.0000e+00

Table A.7: Data associated with the equations of state used in the methane test problem.

k	reaction				frc	frp	frx	rrc	rrp	rrx
1	ch4	=	ch3+h	[1]	1.400e+17	0.00	8.840e+04	2.840e+11	1.00	-1.952e+04
2	ch4+h	=	ch3+h2	[2]	1.250e+14	0.00	1.190e+04	4.800e+12	0.00	1.143e+04
3	ch4+oh	=	ch3+h2o	[3]	3.320e+12	0.00	3.772e+03	5.500e+11	0.00	1.845e+04
4	ch4+o	=	ch3+oh	[4]	2.000e+13	0.00	9.200e+03	3.340e+11	0.00	6.640e+03
5	hco+oh	=	co+h2o	[6]	1.000e+14	0.00	0.	2.800e+15	0.00	1.051e+05
6	oh+co	=	h+co2	[7]	2.520e+11	0.00	1.080e+03	2.835e+13	0.00	2.342e+04
7	h+o2	=	o+oh	[8]	2.200e+14	0.00	1.679e+04	1.740e+13	0.00	6.770e+02
8	h2+o	=	h+oh	[9]	1.800e+10	1.00	8.900e+03	7.890e+09	1.00	7.003e+03
9	o+h2o	=	oh+oh	[10]	6.800e+13	0.00	1.835e+04	6.900e+12	0.00	1.100e+03
10	h+h2o	=	oh+h2	[11]	9.500e+13	0.00	2.030e+04	2.200e+13	0.00	5.146e+03
11	hco	=	h+co	[12]	2.500e+14	0.00	1.900e+04	8.700e+10	1.00	1.553e+03
12	oh	=	o+h	[13]	7.500e+15	0.00	1.022e+05	8.000e+15	0.00	0.
13	o2	=	o+o	[14]	2.500e+19	-1.00	1.190e+05	1.300e+17	-0.75	0.
14	h2	=	h+h	[15]	3.100e+05	0.00	1.040e+05	1.200e+15	0.00	0.
15	o2+h2	=	oh+oh	[16]	7.940e+14	0.00	4.500e+04	2.760e+13	0.00	2.686e+04
16	h2o	=	h+oh	[25]	2.200e+16	0.00	1.050e+05	2.690e+11	1.00	-1.748e+04
17	ho2	=	h+o2	[26]	1.115e+19	-1.00	4.873e+04	1.725e+15	0.00	-1.000e+03
18	co2	=	co+o	[27]	5.500e+21	-1.00	1.318e+05	5.900e+15	0.00	4.093e+03
19	co2+o	=	co+o2	[28]	2.780e+12	0.00	4.383e+04	3.140e+11	0.00	3.760e+04
20	hco+h	=	co+h2	[29]	2.000e+14	0.00	0.	1.310e+15	0.00	9.000e+04
21	o+hco	=	co+oh	[30]	1.000e+14	0.00	0.	2.880e+14	0.00	8.790e+04
22	ch2o	=	hco+h	[31]	5.000e+16	0.00	7.200e+04	2.120e+11	1.00	-2.077e+04
23	ch2o+oh	=	hco+h2o	[32]	5.400e+14	0.00	6.300e+03	1.870e+14	0.00	3.612e+04
24	ch2o+h	=	hco+h2	[33]	1.350e+13	0.00	3.760e+03	1.070e+12	0.00	1.843e+04
25	ch2o+o	=	hco+oh	[34]	5.000e+13	0.00	4.600e+03	1.750e+12	0.00	1.717e+04
26	ch3+oh	=	ch2o+h2	[35]	4.000e+12	0.00	0.	1.200e+14	0.00	7.172e+04
27	ch3+o	=	ch2o+h	[36]	1.300e+14	0.00	2.000e+03	1.700e+15	0.00	7.163e+04
28	ch3+o2	=	ch3o+o	[37]	4.800e+13	0.00	2.900e+04	3.040e+14	0.00	7.330e+02
29	ch2o+ch3	=	hco+ch4	[38]	1.000e+10	0.50	6.000e+03	2.090e+10	0.50	2.114e+04
30	hco+ch3	=	ch4+co	[39]	3.000e+11	0.50	0.	5.140e+13	0.50	9.047e+04
31	ch3o	=	ch2o+h	[40]	5.000e+13	0.00	2.100e+04	9.910e+08	1.00	-2.563e+03
32	ho2	=	o+oh	[41]	8.180e+21	-1.00	6.585e+0	1.000e+17	0.00	0.
33	ho2+o	=	o2+oh	[42]	5.000e+13	0.00	1.000e+03	6.420e+13	0.00	5.661e+04
34	hco+ho2	=	ch2o+o2	[43]	1.000e+14	0.00	3.000e+03	3.660e+15	0.00	4.604e+04
35	ch3o+o2	=	ch2o+ho2	[44]	1.000e+12	0.00	6.000e+03	1.280e+11	0.00	3.217e+04
36	ch3+ho2	=	ch4+o2	[45]	1.000e+12	0.00	4.000e+02	7.630e+13	0.00	5.859e+04
37	hco+o2	=	co+ho2	[46]	3.000e+12	0.00	7.000e+03	6.720e+12	0.00	3.929e+04
38	h+h2o	=	oh+oh	[47]	2.500e+14	0.00	1.900e+03	2.530e+13	0.00	4.140e+04
39	h+h2o	=	h2+o2	[48]	2.500e+13	0.00	7.000e+02	7.310e+13	0.00	5.841e+04
40	oh+h2o	=	h2o+o2	[49]	5.000e+13	0.00	1.000e+03	6.330e+14	0.00	7.386e+04
41	h2o2+o2	=	ho2+ho2	[50]	4.000e+13	0.00	4.264e+04	1.000e+13	0.00	1.000e+03
42	h2o2	=	oh+oh	[51]	1.200e+17	0.00	4.580e+04	4.730e+11	1.00	-6.065e+03
43	ho2+h2	=	h2o2+h	[52]	1.245e+12	0.00	1.988e+04	1.700e+12	0.00	3.800e+03
44	ch4+h2o	=	ch3+h2o2	[53]	2.000e+13	0.00	1.800e+04	1.050e+12	0.00	1.448e+03
45	ch2o+ho2	=	hco+h2o2	[54]	5.000e+12	0.00	6.300e+03	5.450e+11	0.00	4.893e+03

Table A.8: Data associated with the reaction set used in the methane test problem.

k	reaction			irf		irs			
1	ch4	=	ch3+h	[1]	3	11	10	1	0
2	ch4+h	=	ch3+h2	[2]	1	11	1	10	2
3	ch4+oh	=	ch3+h2o	[3]	1	11	5	10	6
4	ch4+o	=	ch3+oh	[4]	1	11	3	10	5
5	hco+oh	=	co+h2o	[6]	1	8	5	7	6
6	oh+co	=	h+co2	[7]	1	5	7	1	9
7	h+o2	=	o+oh	[8]	1	1	4	3	5
8	h2+o	=	h+oh	[9]	1	2	3	1	5
9	o+h2o	=	oh+oh	[10]	2	3	6	5	0
10	h+h2o	=	oh+h2	[11]	1	1	6	5	2
11	hco	=	h+co	[12]	3	8	1	7	0
12	oh	=	o+h	[13]	3	5	3	1	0
13	o2	=	o+o	[14]	4	4	3	0	0
14	h2	=	h+h	[15]	4	2	1	0	0
15	o2+h2	=	oh+oh	[16]	2	4	2	5	0
16	h2o	=	h+oh	[25]	3	6	1	5	0
17	ho2	=	h+o2	[26]	3	12	1	4	0
18	co2	=	co+o	[27]	3	9	7	3	0
19	co2+o	=	co+o2	[28]	1	9	3	7	4
20	hco+h	=	co+h2	[29]	1	8	1	7	2
21	o+hco	=	co+oh	[30]	1	3	8	7	5
22	ch2o	=	hco+h	[31]	3	14	8	1	0
23	ch2o+oh	=	hco+h2o	[32]	1	14	5	8	6
24	ch2o+h	=	hco+h2	[33]	1	14	1	8	2
25	ch2o+o	=	hco+oh	[34]	1	14	3	8	5
26	ch3+oh	=	ch2o+h2	[35]	1	10	5	14	2
27	ch3+o	=	ch2o+h	[36]	1	10	3	14	1
28	ch3+o2	=	ch3o+o	[37]	1	10	4	15	3
29	ch2o+ch3	=	hco+ch4	[38]	1	14	10	8	11
30	hco+ch3	=	ch4+co	[39]	1	8	10	11	7
31	ch3o	=	ch2o+h	[40]	3	15	14	1	0
32	ho2	=	o+oh	[41]	3	12	3	5	0
33	ho2+o	=	o2+oh	[42]	1	12	3	4	5
34	hco+ho2	=	ch2o+o2	[43]	1	8	12	14	4
35	ch3o+o2	=	ch2o+ho2	[44]	1	15	4	14	12
36	ch3+ho2	=	ch4+o2	[45]	1	10	12	11	4
37	hco+o2	=	co+ho2	[46]	1	8	4	7	12
38	h+ho2	=	oh+oh	[47]	2	1	12	5	0
39	h+ho2	=	h2+o2	[48]	1	1	12	2	4
40	oh+ho2	=	h2o+o2	[49]	1	5	12	6	4
41	h2o2+o2	=	ho2+ho2	[50]	2	13	4	12	0
42	h2o2	=	oh+oh	[51]	4	13	5	0	0
43	ho2+h2	=	h2o2+h	[52]	1	12	2	13	1
44	ch4+ho2	=	ch3+h2o2	[53]	1	11	12	10	13
45	ch2o+ho2	=	hco+h2o2	[54]	1	14	12	8	13

Table A.8: Data associated with the reaction set used in the methane test problem: continued.

```

lists = 1:6 12:20

cpa1(2)=7.739 5.365      cpa2(2)=-3.1923 -9.186e-1
cpa3(2)=4.3373 8.955e-1  cpa4(2)=-2.104 -4.39e-1
cpa5(2)=4.8267e-1 1.06e-1 cpa6(2)=-4.3733e-2 -9.8667e-3
h0(3)=5.9553e+1
h0(12)=5.
cpa1(13)=9.36  cpa2(13)=6.193 cpa3(13)=1.943
cpa4(13)=-3.727 cpa5(13)=1.43  cpa6(13)=-1.76e-1
h0(13)=-3.253e1
h0(15)=3.

listr = 1:4 6:16 25:54

rrc(1)=2.84e11 rrp(1)=1. rrx(1)=-1.9515e4
frc(4)=2.e13 frp(4)=0. frx(4)=9.2e3
rrc(4)=3.34e11 rrp(4)=0. rrx(4)=6.64e3
frc(6) = 2.52e11 frp(6) = 1.e-99 frx(6) = 1.08e3
rrc(6) = 2.835e13 rrp(6) = 1.e-99 rrx(6) = 2.342e4
rrc(8)=7.89e9 rrp(8)=1. rrx(8)=7.003e3
rrc(9)=6.9e12
frc(11) = 2.5e14 rrc(11) = 8.7e10
frc(12)=7.5e15 frp(12)=0. frx(12)=1.0222e5 rrc(12)=8.e15
frc(13)=2.5e19 frp(13)=-1. frx(13)=1.19e5
rrc(13)=1.3e17 rrp(13)=-.75 rrx(13)=0.
frc(14)=3.1e5 frp(14)=0. frx(14)=1.04e5
rrc(14)=1.2e15 rrp(14)=0. rrx(14)=0.
frc(15)=7.94e14 frp(15)=0. frx(15)=4.5e4
rrc(15)=2.76e13 rrp(15)=0. rrx(15)=2.6856e4
rrc(16)=2.69e11 rrp(16)=1. rrx(16)=-1.7485e4
frp(17)=-1. frx(17)=4.8733e4
frc(17) = 1.115e19 rrc(17) = 1.725e15
rrx(18)=4.093e3
frc(22)=5.e16 5.4e14 1.35e13 frp(22)=0. 0. 0.
frx(22)=7.2e4 6.3e3 3.76e3
rrc(22)=2.12e11 1.87e14 1.07e12 rrp(22)=1. 0. 0.
rrx(22)=-2.0770e4 3.6123e4 1.843e4
frx(42)=4.58e4 1.988e4
rrc(42)=4.73e11 rrp(42)=1. rrx(42)=-6.065e3
frc(43)=1.245e12 rrx(43)=3.8e3
rrc(44)=1.05e12 rrx(44)=1.448e3

```

Table A.9: Input deck used to generate methane test problem.

```

frx(37)=7.e3 rrx(37)=3.9286e4
rrc(38)=2.53e13 7.31e13
rrp(38)=0. 0.
rrx(38)=4.1399e4 5.8409e4
frc(28) = 4.8e13 rrc(28) = 3.04e14
frc(37) = 3.0e12 rrc(37) = 6.72e12
frc(45) = 5.0e12 frx(45) = 6.3e3
rrc(45) = 5.45e11 rrx(45) = 4.893e3
c = 15(1.e-6) 300. .2 0.
c(4) = .905
c(11) = .095
pr0 = 5.26e4
prbr = 5.26e4
s0 = 1.
s1 = 0.
xsprd = .04
nzones = 25
xmax = 10.0
dzero = 4.58e-7
dczero = 4.16e-7
sorxmax = 0.
sorsprd = .04
sorpcc = 225.
tig = 3.e-4
iprint = 100 0 5 100 100 100
lste4 = 1:25
itimer=1
eta2 = .15
delymn = 300.
euler = 1.
euler2 = -1.
rzflag = 1.
itermx = 20
rdelcmx = .1
dt = 1.e-6
dtmax = 1.
dtmin = 1.e-20
mxcycle=1200

mxcycle = 1990
iprint(3)=10
timemx = .03
nzxcass = 8
$ $

```

Table A.9: Input deck used to generate methane test problem: continued.

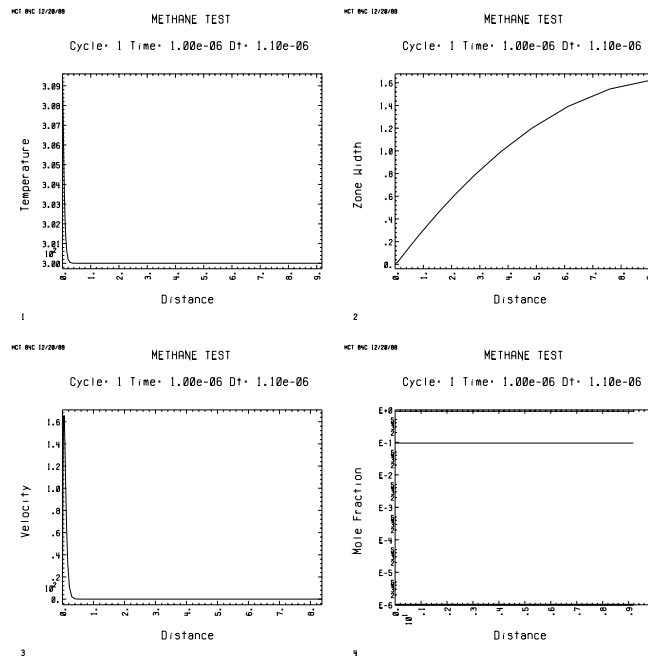


Figure A.11: Methane test problem after initial cycle.

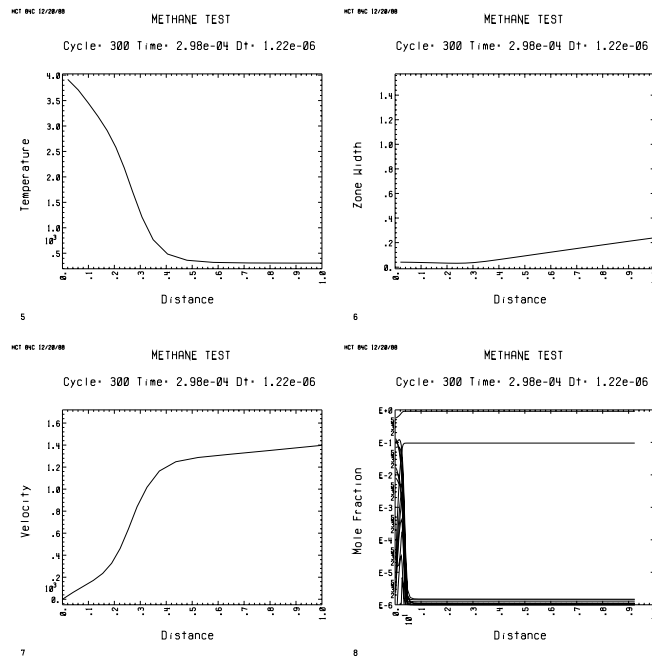


Figure A.12: Left tenth of grid of the methane test problem at early stages of initiation.

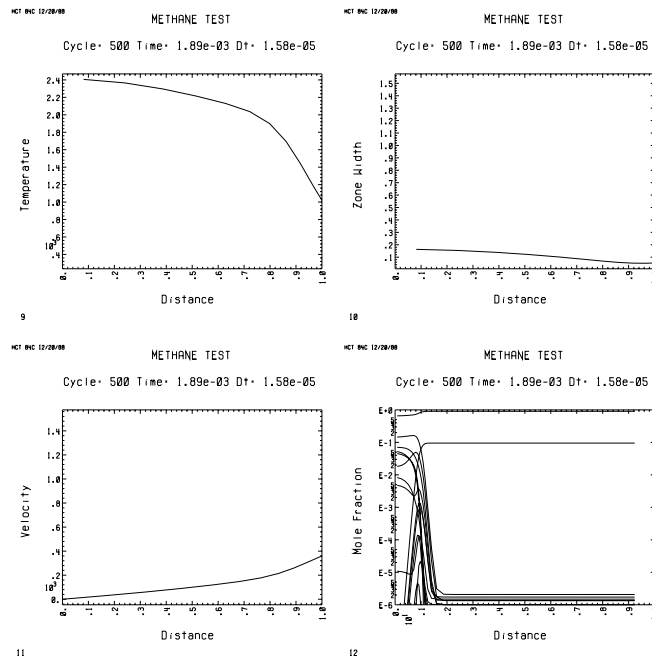


Figure A.13: Left tenth of grid of the methane test problem as flame begins to propagate away from source region.

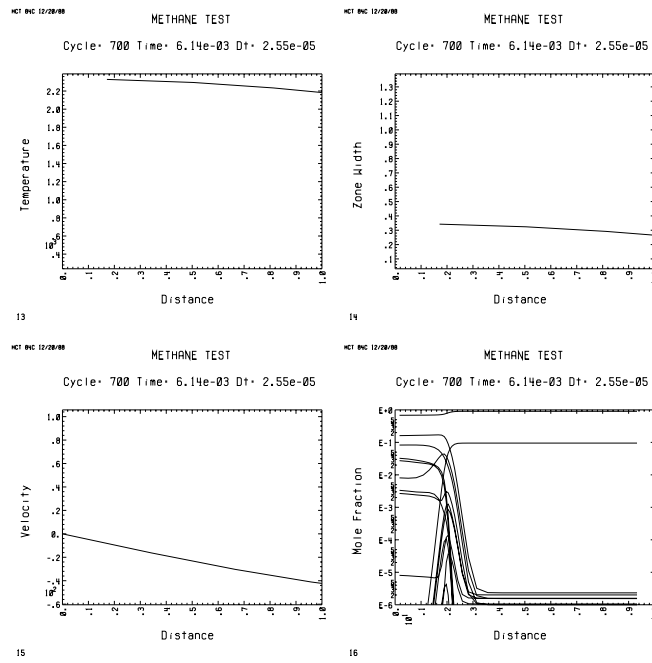


Figure A.14: Methane test problem after propagating flame has moved across most of the grid.

```

                                hct          output                                vers.  84c          12/20/88

cycle=1500 avgiter=  4.71e+00 ircytot=0 irzftot=0 njcbtot=1019 time=2.9419e-02 dt=3.14e-05

species limiting dt- 13 zone- 13
slowest converging species-  1 zone- 17

jd          r          dr          ctot          rho          t          u          press          edotr          edotc          edotd          p*dv/dx          vsound

 1          0.  1.26e+00  2.81e-07  8.44e-06  2248.96          0.  5.26e+04  2.69e+04  -1.77e+05  -5.09e+03  3.53e+04  8.81e+04
 2  1.2553e+00  1.18e+00  2.85e-07  8.57e-06  2219.50  -8.43e-01  5.26e+04  6.71e+04  -7.29e+04  -2.15e+02  3.32e+03  8.74e+04
 3  2.4397e+00  1.06e+00  2.90e-07  8.74e-06  2181.56  -9.18e-01  5.26e+04  1.79e+05  2.26e+04  3.70e+03  -3.87e+04  8.65e+04
 4  3.4996e+00  9.05e-01  2.94e-07  8.87e-06  2150.67  -1.39e-01  5.26e+04  4.35e+05  9.10e+03  5.77e+03  -8.13e+04  8.59e+04
 5  4.4043e+00  7.41e-01  2.98e-07  8.97e-06  2125.35  1.26e+00  5.26e+04  9.45e+05  -1.31e+05  8.50e+03  -1.42e+05  8.55e+04
 6  5.1449e+00  5.86e-01  3.02e-07  9.07e-06  2097.48  3.27e+00  5.26e+04  1.84e+06  -3.92e+05  1.45e+04  -2.50e+05  8.50e+04
 7  5.7306e+00  4.51e-01  3.07e-07  9.20e-06  2061.88  6.05e+00  5.26e+04  3.23e+06  -7.84e+05  2.44e+04  -4.21e+05  8.45e+04
 8  6.1816e+00  3.41e-01  3.13e-07  9.37e-06  2018.04  9.65e+00  5.26e+04  4.96e+06  -1.22e+06  3.46e+04  -6.46e+05  8.38e+04
 9  6.5223e+00  2.54e-01  3.21e-07  9.56e-06  1969.87  1.38e+01  5.26e+04  6.66e+06  -1.53e+06  3.42e+04  -8.93e+05  8.30e+04
10  6.7764e+00  1.88e-01  3.29e-07  9.76e-06  1923.03  1.82e+01  5.26e+04  1.25e+07  -5.01e+06  -1.21e+04  -1.36e+06  8.22e+04
11  6.9642e+00  1.38e-01  3.38e-07  1.00e-05  1868.82  2.30e+01  5.26e+04  5.30e+07  -3.45e+07  -3.47e+05  -3.64e+06  8.12e+04
12  7.1024e+00  1.03e-01  3.61e-07  1.07e-05  1754.16  3.26e+01  5.26e+04  1.38e+08  -9.06e+07  -1.31e+06  -9.72e+06  7.88e+04
13  7.2056e+00  8.20e-02  4.10e-07  1.22e-05  1544.17  5.16e+01  5.26e+04  1.40e+08  -6.19e+07  -1.59e+06  -1.65e+07  7.39e+04
14  7.2877e+00  7.33e-02  4.89e-07  1.46e-05  1292.86  7.74e+01  5.26e+04  5.55e+07  3.15e+07  -8.06e+05  -1.90e+07  6.76e+04
15  7.3609e+00  7.36e-02  6.00e-07  1.80e-05  1053.50  1.04e+02  5.26e+04  1.04e+07  6.72e+07  -2.05e+05  -1.73e+07  6.11e+04
16  7.4345e+00  8.12e-02  7.56e-07  2.28e-05  836.08  1.28e+02  5.26e+04  1.41e+06  6.00e+07  9.03e+04  -1.43e+07  5.46e+04
17  7.5157e+00  9.69e-02  9.92e-07  3.00e-05  637.86  1.50e+02  5.26e+04  1.62e+05  4.38e+07  1.61e+05  -1.07e+07  4.80e+04
18  7.6126e+00  1.23e-01  1.35e-06  4.12e-05  467.07  1.70e+02  5.26e+04  1.10e+04  2.66e+07  9.98e+04  -6.93e+06  4.16e+04
19  7.7357e+00  1.62e-01  1.78e-06  5.43e-05  354.65  1.86e+02  5.26e+04  1.36e+03  1.06e+07  3.18e+04  -2.96e+06  3.67e+04
20  7.8973e+00  2.11e-01  2.02e-06  6.15e-05  313.15  1.95e+02  5.26e+04  2.44e+02  2.12e+06  5.24e+03  -6.13e+05  3.46e+04
21  8.1087e+00  2.70e-01  2.09e-06  6.36e-05  302.89  1.98e+02  5.26e+04  9.95e+01  3.05e+05  6.00e+02  -8.69e+04  3.41e+04
22  8.3782e+00  3.32e-01  2.10e-06  6.41e-05  300.71  1.98e+02  5.26e+04  7.86e+01  4.08e+04  4.21e+01  -9.96e+03  3.40e+04
23  8.7101e+00  3.91e-01  2.11e-06  6.42e-05  300.23  1.98e+02  5.26e+04  7.50e+01  6.10e+03  7.21e-01  -3.85e+02  3.40e+04
24  9.1015e+00  4.39e-01  2.11e-06  6.42e-05  300.13  1.98e+02  5.26e+04  7.43e+01  9.53e+02  -1.24e-01  3.15e+02  3.40e+04
25  9.5407e+00  4.66e-01  2.11e-06  6.42e-05  300.09  1.98e+02  5.26e+04  7.41e+01  3.63e+02  -3.18e-02  -6.17e+02  3.40e+04
                                total energy production rate  4.77e+07  1.71e-07  -2.99e+05  -1.04e+07

total mass=  2.207584e-04 total internal energy= -9.594495e+05

time spent (sec)-cpu= 1.597e+02 i-o= 1.253e+01 sys= 4.604e+00

subroutine  sec/call  sec/cycle
diffun     7.57e-03   3.05e-02
difped     2.51e-02   1.71e-02
solbtl     2.01e-03   9.46e-03
decbtl     3.12e-02   2.12e-02
rezone     5.47e-04   5.47e-04
ckenvg     6.77e-04   3.19e-03
remap      2.18e-03   2.17e-03

```

Table A.10: General Lagrange edit of methane test problem at cycle 1500.

specie mole fractions

zone	h	h2	o	o2	oh	h2o	co
1	1.25e-03	1.40e-03	1.36e-02	6.97e-01	2.20e-02	1.72e-01	3.80e-03
2	1.00e-03	1.20e-03	1.15e-02	7.00e-01	1.99e-02	1.74e-01	3.20e-03
3	7.82e-04	1.01e-03	9.57e-03	7.02e-01	1.76e-02	1.76e-01	2.64e-03
4	7.25e-04	9.54e-04	8.93e-03	7.02e-01	1.66e-02	1.77e-01	2.49e-03
5	8.30e-04	1.03e-03	9.63e-03	7.01e-01	1.68e-02	1.77e-01	2.77e-03
6	1.09e-03	1.22e-03	1.14e-02	6.99e-01	1.78e-02	1.76e-01	3.52e-03
7	1.52e-03	1.49e-03	1.39e-02	6.96e-01	1.89e-02	1.75e-01	5.05e-03
8	2.02e-03	1.75e-03	1.63e-02	6.93e-01	1.94e-02	1.74e-01	8.28e-03
9	2.29e-03	1.85e-03	1.72e-02	6.93e-01	1.89e-02	1.73e-01	1.47e-02
10	2.01e-03	1.68e-03	1.53e-02	6.97e-01	1.68e-02	1.74e-01	2.52e-02
11	1.30e-03	1.46e-03	1.06e-02	7.07e-01	1.31e-02	1.73e-01	3.71e-02
12	7.31e-04	1.90e-03	5.54e-03	7.27e-01	8.00e-03	1.63e-01	4.36e-02
13	4.14e-04	2.81e-03	2.00e-03	7.59e-01	3.26e-03	1.37e-01	4.01e-02
14	1.85e-04	2.90e-03	4.62e-04	7.94e-01	8.45e-04	1.05e-01	3.14e-02
15	5.92e-05	2.28e-03	7.21e-05	8.25e-01	1.48e-04	7.55e-02	2.26e-02
16	9.79e-06	1.54e-03	7.26e-06	8.52e-01	1.60e-05	5.03e-02	1.51e-02
17	6.24e-08	9.01e-04	5.84e-07	8.73e-01	1.03e-06	2.94e-02	8.82e-03
18	1.09e-10	4.18e-04	5.33e-08	8.90e-01	5.35e-08	1.36e-02	4.08e-03
19	6.93e-12	1.36e-04	8.81e-09	9.00e-01	2.79e-09	4.39e-03	1.32e-03
20	6.49e-12	2.95e-05	1.38e-08	9.04e-01	5.04e-10	9.30e-04	2.82e-04
21	6.76e-12	5.06e-06	2.08e-08	9.05e-01	6.17e-10	1.23e-04	3.87e-05
22	6.14e-12	1.65e-06	2.36e-08	9.05e-01	7.07e-10	1.41e-05	5.87e-06
23	5.94e-12	1.15e-06	2.43e-08	9.05e-01	7.35e-10	4.18e-06	2.48e-06
24	5.90e-12	1.05e-06	2.45e-08	9.05e-01	7.43e-10	2.85e-06	1.93e-06
25	5.89e-12	1.02e-06	2.46e-08	9.05e-01	7.45e-10	2.54e-06	1.79e-06
avg	3.35e-04	4.54e-04	3.43e-03	8.42e-01	5.43e-03	5.42e-02	2.65e-03
zone	hco	co2	ch3	ch4	ho2	h2o2	ch2o
1	5.16e-11	8.84e-02	3.69e-20	1.85e-21	5.78e-06	1.33e-07	1.78e-13
2	3.61e-11	8.94e-02	2.25e-20	3.29e-20	5.30e-06	1.27e-07	1.28e-13
3	2.38e-11	9.04e-02	7.46e-19	1.10e-17	4.79e-06	1.22e-07	8.60e-14
4	1.98e-11	9.08e-02	1.66e-16	2.53e-15	4.56e-06	1.28e-07	6.93e-14
5	2.20e-11	9.07e-02	2.50e-14	3.92e-13	4.65e-06	1.52e-07	8.03e-14
6	3.15e-11	9.00e-02	2.60e-12	4.27e-11	5.02e-06	2.00e-07	1.22e-12
7	9.04e-11	8.84e-02	1.94e-10	3.34e-09	5.58e-06	2.79e-07	9.67e-11
8	2.32e-09	8.51e-02	1.01e-08	1.85e-07	6.13e-06	3.91e-07	5.80e-09
9	8.28e-08	7.87e-02	3.55e-07	6.78e-06	6.36e-06	5.06e-07	2.23e-07
10	1.69e-06	6.83e-02	7.46e-06	1.48e-04	6.30e-06	5.54e-07	4.78e-06
11	1.61e-05	5.49e-02	8.13e-05	1.65e-03	9.69e-06	5.23e-07	4.88e-05
12	6.28e-05	4.17e-02	4.03e-04	8.55e-03	3.42e-05	1.36e-06	2.29e-04
13	1.02e-04	3.09e-02	9.41e-04	2.28e-02	9.67e-05	1.35e-05	5.65e-04
14	7.11e-05	2.26e-02	1.24e-03	3.98e-02	1.42e-04	3.85e-05	8.15e-04
15	2.17e-05	1.61e-02	1.08e-03	5.57e-02	1.10e-04	4.21e-05	7.45e-04
16	3.39e-06	1.07e-02	7.05e-04	6.94e-02	6.22e-05	3.27e-05	5.27e-04
17	3.72e-07	6.24e-03	3.97e-04	8.06e-02	2.88e-05	2.04e-05	3.12e-04
18	4.47e-08	2.88e-03	1.81e-04	8.89e-02	1.09e-05	1.00e-05	1.46e-04
19	8.84e-09	9.26e-04	5.89e-05	9.33e-02	3.69e-06	3.97e-06	4.84e-05
20	1.60e-08	1.95e-04	1.38e-05	9.46e-02	1.80e-06	1.84e-06	1.16e-05
21	2.59e-08	2.60e-05	3.89e-06	9.49e-02	1.46e-06	1.38e-06	3.48e-06
22	3.01e-08	3.65e-06	2.35e-06	9.50e-02	1.41e-06	1.29e-06	2.17e-06
23	3.14e-08	1.45e-06	2.07e-06	9.50e-02	1.40e-06	1.27e-06	1.92e-06
24	3.17e-08	1.10e-06	2.01e-06	9.50e-02	1.40e-06	1.26e-06	1.87e-06
25	3.18e-08	1.02e-06	2.00e-06	9.50e-02	1.40e-06	1.26e-06	1.86e-06
avg	1.45e-06	2.61e-02	3.94e-05	6.57e-02	5.71e-06	2.32e-06	2.84e-05

Table A.11: Edit of species mole fractions in methane test problem at cycle 1500.

specie mole fractions

zone	ch3o
1	5.48e-22
2	3.60e-22
3	1.08e-20
4	2.32e-18
5	3.04e-16
6	2.53e-14
7	1.41e-12
8	5.55e-11
9	1.57e-09
10	3.14e-08
11	3.92e-07
12	2.17e-06
13	4.22e-06
14	2.96e-06
15	7.79e-07
16	1.07e-07
17	1.32e-08
18	4.66e-09
19	1.64e-08
20	3.00e-08
21	3.61e-08
22	3.78e-08
23	3.84e-08
24	3.85e-08
25	3.86e-08
avg	7.71e-08

total moles in problem- 7.2783e-06

Table A.11: Edit of species mole fractions in methane test problem at cycle 1500: continued.

zone 13	reaction	rate	h	h2	o
	(concentration)		1.70e-10	1.15e-09	8.19e-10
	total rate		1.14e-04	1.95e-05	1.23e-04
	net rate		1.63e-07	-7.92e-07	1.47e-06
	diffusion		1.25e-03	-9.43e-02	3.32e-02
	hydrodynamics	-3.142e+02	-4.66e-04	-1.86e-02	-2.09e-03
	chemistry		6.49e-04	7.21e-02	-1.92e-02
7	h+o2=o+oh	4.877e-05	-4.27e-01	0.	3.96e-01
27	ch3+o=ch2o+h	2.125e-05	1.86e-01	0.	-1.73e-01
-7	o+oh=h+o2	1.524e-05	1.33e-01	0.	-1.24e-01
11	hco=h+co	8.868e-06	7.76e-02	0.	0.
-9	oh+oh=o+h2o	8.583e-06	0.	0.	6.98e-02
9	o+h2o=oh+oh	7.893e-06	0.	0.	-6.42e-02
4	ch4+o=ch3+oh	7.619e-06	0.	0.	-6.19e-02
-10	oh+h2=h+h2o	6.314e-06	5.52e-02	-3.25e-01	0.
2	ch4+h=ch3+h2	4.090e-06	-3.58e-02	2.10e-01	0.
6	oh+co=h+co2	3.887e-06	3.40e-02	0.	0.
21	o+hco=co+oh	3.402e-06	0.	0.	-2.77e-02
25	ch2o+o=hco+oh	2.116e-06	0.	0.	-1.72e-02
26	ch3+oh=ch2o+h2	2.057e-06	0.	1.06e-01	0.
8	h2+o=h+oh	1.441e-06	1.26e-02	-7.41e-02	-1.17e-02
20	hco+h=co+h2	1.395e-06	-1.22e-02	7.17e-02	0.
10	h+h2o=oh+h2	1.210e-06	-1.06e-02	6.22e-02	0.
-8	h+oh=h2+o	2.815e-07	-2.46e-03	1.45e-02	2.29e-03
zone 13	reaction	rate	o2	oh	h2o
	(concentration)		3.11e-07	1.33e-09	5.61e-08
	total rate		1.80e-04	1.75e-04	1.04e-04
	net rate		-1.22e-04	2.19e-06	7.44e-06
	diffusion		5.64e-02	2.64e-02	-2.07e-01
	hydrodynamics	-3.142e+02	-5.42e-01	-2.40e-03	-1.69e-01
	chemistry		-1.92e-01	-1.15e-02	4.48e-01
7	h+o2=o+oh	4.877e-05	-2.71e-01	2.79e-01	0.
23	ch2o+oh=hco+h2o	2.126e-05	0.	-1.22e-01	2.05e-01
-7	o+oh=h+o2	1.524e-05	8.45e-02	-8.72e-02	0.
3	ch4+oh=ch3+h2o	1.208e-05	0.	-6.92e-02	1.16e-01
-9	oh+oh=o+h2o	8.583e-06	0.	-9.83e-02	8.26e-02
9	o+h2o=oh+oh	7.893e-06	0.	9.04e-02	-7.59e-02
4	ch4+o=ch3+oh	7.619e-06	0.	4.36e-02	0.
-10	oh+h2=h+h2o	6.314e-06	0.	-3.61e-02	6.07e-02
5	hco+oh=co+h2o	5.542e-06	0.	-3.17e-02	5.33e-02
37	hco+o2=co+ho2	3.996e-06	-2.22e-02	0.	0.
6	oh+co=h+co2	3.887e-06	0.	-2.22e-02	0.
21	o+hco=co+oh	3.402e-06	0.	1.95e-02	0.
25	ch2o+o=hco+oh	2.116e-06	0.	1.21e-02	0.
26	ch3+oh=ch2o+h2	2.057e-06	0.	-1.18e-02	0.
40	oh+ho2=h2o+o2	1.902e-06	1.05e-02	-1.09e-02	1.83e-02
10	h+h2o=oh+h2	1.210e-06	0.	6.92e-03	-1.16e-02
38	h+ho2=oh+oh	8.928e-07	0.	1.02e-02	0.

Table A.12: Lagrange edit of rates affecting physical variables in zone 13 for methane test problem at cycle 1500.

zone 13	reaction	rate	co	hco	co2
	(concentration)		1.64e-08	4.19e-11	1.26e-08
	total rate		4.62e-05	4.73e-05	1.16e-05
	net rate		6.00e-07	-1.44e-08	3.62e-06
	diffusion		-2.97e-01	-3.48e-03	3.20e-01
	hydrodynamics	-3.142e+02	-1.12e-01	-2.79e-04	-3.42e-01
	chemistry		4.22e-01	3.46e-03	3.33e-01
23	ch2o+oh=hco+h2o	2.126e-05	0.	4.50e-01	0.
11	hco=h+co	8.868e-06	1.92e-01	-1.88e-01	0.
5	hco+oh=co+h2o	5.542e-06	1.20e-01	-1.17e-01	0.
37	hco+o2=co+ho2	3.996e-06	8.64e-02	-8.46e-02	0.
6	oh+co=h+co2	3.887e-06	-8.40e-02	0.	3.34e-01
21	o+hco=co+oh	3.402e-06	7.36e-02	-7.20e-02	0.
25	ch2o+o=hco+oh	2.116e-06	0.	4.48e-02	0.
20	hco+h=co+h2	1.395e-06	3.02e-02	-2.95e-02	0.
zone 13	reaction	rate	ch3	ch4	ho2
	(concentration)		3.86e-10	9.33e-09	3.96e-11
	total rate		4.88e-05	3.70e-05	8.57e-06
	net rate		-5.21e-07	-1.65e-05	-6.48e-08
	diffusion		-9.53e-03	2.69e-01	-3.38e-03
	hydrodynamics	-3.142e+02	-2.48e-03	-7.92e-02	-1.45e-03
	chemistry		1.33e-03	-6.35e-01	-2.73e-03
27	ch3+o=ch2o+h	2.125e-05	-4.35e-01	0.	0.
3	ch4+oh=ch3+h2o	1.208e-05	2.48e-01	-3.26e-01	0.
4	ch4+o=ch3+oh	7.619e-06	1.56e-01	-2.06e-01	0.
2	ch4+h=ch3+h2	4.090e-06	8.38e-02	-1.11e-01	0.
37	hco+o2=co+ho2	3.996e-06	0.	0.	4.66e-01
26	ch3+oh=ch2o+h2	2.057e-06	-4.21e-02	0.	0.
40	oh+ho2=h2o+o2	1.902e-06	0.	0.	-2.22e-01
33	ho2+o=o2+oh	1.167e-06	0.	0.	-1.36e-01
38	h+ho2=oh+oh	8.928e-07	0.	0.	-1.04e-01
39	h+ho2=h2+o2	1.335e-07	0.	0.	-1.56e-02

Table A.12: Lagrange edit of rates affecting physical variables in zone 13 for methane test problem at cycle 1500: continued.

zone 13	reaction	rate	h2o2	ch2o	ch3o
	(concentration)		5.51e-12	2.32e-10	1.73e-12
	total rate		1.70e-07	4.73e-05	9.05e-07
	net rate		-1.74e-08	-3.47e-07	-8.65e-10
	diffusion		2.01e-01	-2.99e-03	-8.41e-03
	hydrodynamics	-3.142e+02	-1.02e-02	-1.54e-03	-5.99e-04
	chemistry		-2.93e-01	-2.81e-03	8.06e-03
23	ch2o+oh=hco+h2o	2.126e-05	0.	-4.49e-01	0.
27	ch3+o=ch2o+h	2.125e-05	0.	4.49e-01	0.
25	ch2o+o=hco+oh	2.116e-06	0.	-4.47e-02	0.
26	ch3+oh=ch2o+h2	2.057e-06	0.	4.35e-02	0.
28	ch3+o2=ch3o+o	4.521e-07	0.	0.	5.00e-01
-28	ch3o+o=ch3+o2	3.308e-07	0.	0.	-3.65e-01
42	h2o2=oh+oh	9.004e-08	-5.30e-01	0.	0.
35	ch3o+o2=ch2o+ho2	7.597e-08	0.	1.61e-03	-8.39e-02
31	ch3o=ch2o+h	3.805e-08	0.	8.04e-04	-4.20e-02
44	ch4+ho2=ch3+h2o2	2.093e-08	1.23e-01	0.	0.
-41	ho2+ho2=h2o2+o2	1.132e-08	6.66e-02	0.	0.
45	ch2o+ho2=hco+h2o2	5.885e-09	3.46e-02	-1.24e-04	0.
-42	oh+oh=h2o2	3.884e-09	2.28e-02	0.	0.
zone 13	reaction	rate	temp		
	(concentration)		1544.17		
	total rate		2.43e+06		
	net rate		4.63e+05		
	conduction		-1.97e-01		
	diffusion		-5.06e-03		
	hydrodynamics	-3.142e+02	-5.27e-02		
	chemistry		4.46e-01		
7	h+o2=o+oh	4.877e-05	-1.05e-01		
23	ch2o+oh=hco+h2o	2.126e-05	8.45e-02		
27	ch3+o=ch2o+h	2.125e-05	1.97e-01		
-7	o+oh=h+o2	1.524e-05	3.27e-02		
3	ch4+oh=ch3+h2o	1.208e-05	2.36e-02		
11	hco=h+co	8.868e-06	-1.71e-02		
-9	oh+oh=o+h2o	8.583e-06	1.98e-02		
9	o+h2o=oh+oh	7.893e-06	-1.82e-02		
-10	oh+h2=h+h2o	6.314e-06	1.28e-02		
5	hco+oh=co+h2o	5.542e-06	7.77e-02		
37	hco+o2=co+ho2	3.996e-06	1.72e-02		
6	oh+co=h+co2	3.887e-06	1.15e-02		
21	o+hco=co+oh	3.402e-06	3.99e-02		
26	ch3+oh=ch2o+h2	2.057e-06	1.97e-02		
40	oh+ho2=h2o+o2	1.902e-06	1.85e-02		
20	hco+h=co+h2	1.395e-06	1.67e-02		

Table A.12: Lagrange edit of rates affecting physical variables in zone 13 for methane test problem at cycle 1500: continued.

lagrange relative change per cycle

zone	h	h2	o	o2	oh	h2o	co
1	-3.43e-04	-2.26e-04	-2.45e-04	2.31e-05	-1.38e-04	3.73e-05	-2.62e-04
2	-1.99e-04	-1.28e-04	-1.34e-04	2.50e-06	-6.33e-05	8.69e-06	-1.54e-04
3	-3.10e-04	-1.90e-04	-1.87e-04	-2.27e-05	-5.45e-05	-2.01e-05	-2.46e-04
4	-1.07e-03	-6.72e-04	-6.69e-04	-4.59e-05	-2.42e-04	-3.48e-05	-8.44e-04
5	-2.46e-03	-1.55e-03	-1.56e-03	-7.82e-05	-5.95e-04	-4.78e-05	-2.03e-03
6	-4.06e-03	-2.53e-03	-2.55e-03	-1.39e-04	-9.38e-04	-8.26e-05	-3.99e-03
7	-5.17e-03	-3.14e-03	-3.17e-03	-2.52e-04	-1.01e-03	-1.69e-04	-7.25e-03
8	-4.52e-03	-2.57e-03	-2.57e-03	-4.44e-04	-4.17e-04	-3.46e-04	-1.17e-02
9	-9.48e-05	2.95e-04	6.04e-04	-7.41e-04	1.47e-03	-6.43e-04	-1.57e-02
10	9.89e-03	3.68e-03	8.21e-03	-1.32e-03	5.85e-03	-9.25e-04	-1.72e-02
11	2.12e-02	-6.75e-03	2.13e-02	-3.31e-03	1.54e-02	-5.18e-04	-1.42e-02
12	2.54e-02	-2.60e-02	3.97e-02	-7.92e-03	3.33e-02	1.46e-03	-6.72e-03
13	3.16e-02	-2.16e-02	6.03e-02	-1.24e-02	5.50e-02	4.26e-03	1.17e-03
14	4.29e-02	-4.55e-03	7.27e-02	-1.35e-02	6.83e-02	6.18e-03	5.57e-03
15	5.20e-02	4.57e-03	7.18e-02	-1.18e-02	6.92e-02	7.02e-03	6.97e-03
16	6.32e-02	7.09e-03	6.09e-02	-9.40e-03	6.16e-02	7.38e-03	7.36e-03
17	7.32e-02	7.57e-03	4.17e-02	-6.88e-03	4.61e-02	7.66e-03	7.63e-03
18	3.60e-02	7.82e-03	2.34e-02	-4.34e-03	3.01e-02	7.91e-03	7.87e-03
19	1.03e-02	8.17e-03	5.15e-03	-1.82e-03	1.91e-02	8.30e-03	8.25e-03
20	-4.34e-03	8.62e-03	-6.40e-03	-3.78e-04	-2.44e-03	8.88e-03	8.81e-03
21	-1.43e-03	7.26e-03	-2.99e-03	-5.36e-05	-3.98e-03	9.86e-03	9.46e-03
22	-1.79e-03	1.97e-03	-2.15e-03	-6.10e-06	-2.94e-03	7.73e-03	5.54e-03
23	-1.94e-03	2.73e-04	-1.98e-03	-2.37e-07	-2.72e-03	1.60e-03	9.28e-04
24	-1.95e-03	4.40e-05	-1.95e-03	1.89e-07	-2.68e-03	2.28e-04	1.69e-04
25	-1.95e-03	1.27e-05	-1.94e-03	-3.71e-07	-2.67e-03	7.35e-05	7.88e-05
zone	hco	co2	ch3	ch4	ho2	h2o2	ch2o
1	-5.27e-04	3.60e-05	-8.67e-04	-2.50e-03	-1.14e-04	-8.95e-05	-4.69e-04
2	-2.29e-04	9.50e-06	-4.09e-03	-2.81e-02	-5.63e-05	-1.23e-04	-1.63e-04
3	-1.80e-04	-1.61e-05	-2.96e-02	-3.00e-02	-5.77e-05	-3.77e-04	7.81e-06
4	-9.69e-04	-2.67e-05	-3.15e-02	-3.17e-02	-2.56e-04	-1.09e-03	-6.31e-04
5	-2.85e-03	-2.57e-05	-3.42e-02	-3.44e-02	-7.18e-04	-2.33e-03	-8.91e-03
6	-7.94e-03	2.28e-06	-3.84e-02	-3.87e-02	-1.38e-03	-3.96e-03	-3.70e-02
7	-2.85e-02	1.62e-04	-4.47e-02	-4.51e-02	-1.91e-03	-5.63e-03	-4.57e-02
8	-5.26e-02	7.71e-04	-5.33e-02	-5.37e-02	-1.74e-03	-6.51e-03	-5.43e-02
9	-6.41e-02	2.43e-03	-6.40e-02	-6.45e-02	-7.74e-04	-5.03e-03	-6.46e-02
10	-7.35e-02	5.63e-03	-7.51e-02	-7.56e-02	-8.87e-03	-8.92e-04	-7.45e-02
11	-7.32e-02	9.32e-03	-7.95e-02	-8.06e-02	-4.52e-02	-2.54e-02	-7.77e-02
12	-5.23e-02	1.06e-02	-6.76e-02	-7.20e-02	-6.82e-02	-9.20e-02	-6.83e-02
13	-1.09e-02	9.22e-03	-4.15e-02	-5.37e-02	-4.99e-02	-9.26e-02	-4.59e-02
14	3.24e-02	7.75e-03	-1.44e-02	-3.58e-02	-1.40e-02	-3.98e-02	-1.84e-02
15	5.47e-02	7.35e-03	2.90e-03	-2.29e-02	9.17e-03	-6.54e-03	-1.11e-04
16	5.32e-02	7.46e-03	8.56e-03	-1.47e-02	1.39e-02	3.42e-03	6.03e-03
17	3.76e-02	7.71e-03	8.26e-03	-9.18e-03	1.25e-02	6.23e-03	7.32e-03
18	2.10e-02	7.97e-03	7.96e-03	-5.17e-03	9.52e-03	6.54e-03	7.66e-03
19	3.37e-03	8.38e-03	8.10e-03	-2.01e-03	6.27e-03	5.76e-03	7.94e-03
20	-7.41e-03	9.00e-03	7.81e-03	-4.04e-04	1.64e-03	2.68e-03	7.57e-03
21	-3.63e-03	9.78e-03	3.67e-03	-5.66e-05	-1.03e-04	6.08e-04	3.47e-03
22	-2.63e-03	6.08e-03	5.35e-04	-6.47e-06	-2.98e-04	2.36e-04	6.05e-04
23	-2.42e-03	1.00e-03	2.20e-05	-2.86e-07	-3.12e-04	1.89e-04	1.39e-04
24	-2.38e-03	1.43e-04	-4.76e-05	1.82e-07	-3.14e-04	1.82e-04	7.51e-05
25	-2.37e-03	4.03e-05	-5.75e-05	-3.72e-07	-3.14e-04	1.80e-04	6.61e-05

Table A.13: Edit of the relative changes in the physical variables during the Lagrange part of cycle 1500 of the methane test problem.

lagrange relative change per cycle

zone	ch3o	t	dx	u
1	-7.14e-04	-1.75e-05	-2.12e-05	-4.46e-01
2	-3.58e-03	-3.86e-07	-1.99e-06	-9.49e-03
3	-2.92e-02	2.45e-05	2.32e-05	-1.60e-02
4	-3.09e-02	5.42e-05	4.88e-05	-8.41e-02
5	-3.29e-02	1.00e-04	8.56e-05	7.45e-03
6	-3.66e-02	1.80e-04	1.50e-04	2.07e-03
7	-4.24e-02	3.05e-04	2.53e-04	-6.78e-04
8	-5.10e-02	4.70e-04	3.88e-04	-2.53e-03
9	-6.26e-02	6.49e-04	5.36e-04	-3.48e-03
10	-7.62e-02	9.41e-04	8.20e-04	-4.46e-03
11	-8.31e-02	2.25e-03	2.19e-03	-9.68e-03
12	-6.39e-02	5.72e-03	5.87e-03	-2.10e-02
13	-1.58e-02	9.58e-03	1.00e-02	-2.51e-02
14	3.57e-02	1.10e-02	1.15e-02	-1.95e-02
15	5.98e-02	1.02e-02	1.05e-02	-1.17e-02
16	5.36e-02	8.44e-03	8.66e-03	-5.65e-03
17	3.00e-02	6.41e-03	6.51e-03	-1.61e-03
18	-4.73e-03	4.19e-03	4.20e-03	1.12e-03
19	-1.29e-02	1.79e-03	1.78e-03	2.76e-03
20	-5.12e-03	3.66e-04	3.70e-04	2.90e-03
21	-3.19e-03	5.38e-05	5.24e-05	2.69e-03
22	-2.80e-03	7.64e-06	6.01e-06	2.78e-03
23	-2.74e-03	1.43e-06	2.32e-07	2.93e-03
24	-2.72e-03	3.31e-07	-1.90e-07	3.07e-03
25	-2.72e-03	-4.43e-08	3.72e-07	3.18e-03

Table A.13: Edit of the relative changes in the physical variables during the Lagrange part of cycle 1500 of the methane test problem: continued.

remap relative change per cycle

zone	h	h2	o	o2	oh	h2o	co
1	-8.17e-05	-5.46e-05	-5.94e-05	6.46e-06	-3.49e-05	9.94e-06	-6.25e-05
2	-2.70e-04	-1.82e-04	-1.99e-04	2.31e-05	-1.21e-04	3.49e-05	-2.06e-04
3	-2.71e-04	-1.88e-04	-2.13e-04	3.68e-05	-1.48e-04	5.30e-05	-2.04e-04
4	2.27e-04	1.31e-04	1.09e-04	4.15e-05	-9.17e-06	5.18e-05	1.90e-04
5	1.21e-03	7.60e-04	7.48e-04	5.20e-05	2.62e-04	4.85e-05	1.03e-03
6	2.47e-03	1.54e-03	1.54e-03	8.95e-05	5.52e-04	6.60e-05	2.50e-03
7	3.53e-03	2.14e-03	2.15e-03	1.75e-04	6.74e-04	1.29e-04	5.07e-03
8	3.32e-03	1.88e-03	1.87e-03	3.33e-04	2.82e-04	2.72e-04	8.84e-03
9	-5.35e-05	-2.80e-04	-5.89e-04	5.93e-04	-1.23e-03	5.27e-04	1.26e-02
10	-8.41e-03	-2.83e-03	-7.08e-03	1.13e-03	-5.04e-03	7.84e-04	1.45e-02
11	-1.83e-02	6.55e-03	-1.88e-02	2.97e-03	-1.37e-02	4.37e-04	1.24e-02
12	-2.29e-02	2.28e-02	-3.55e-02	7.16e-03	-2.99e-02	-1.33e-03	6.06e-03
13	-2.90e-02	1.94e-02	-5.40e-02	1.12e-02	-4.93e-02	-3.84e-03	-9.95e-04
14	-3.84e-02	4.30e-03	-6.41e-02	1.21e-02	-6.04e-02	-5.54e-03	-4.97e-03
15	-4.54e-02	-3.90e-03	-6.10e-02	1.03e-02	-5.90e-02	-6.12e-03	-6.08e-03
16	-4.76e-02	-5.77e-03	-4.70e-02	7.69e-03	-4.75e-02	-6.03e-03	-6.02e-03
17	-3.49e-02	-5.32e-03	-2.77e-02	4.82e-03	-2.98e-02	-5.37e-03	-5.35e-03
18	-1.41e-02	-3.48e-03	-1.07e-02	2.03e-03	-1.25e-02	-3.51e-03	-3.50e-03
19	-2.46e-03	-2.02e-04	-2.14e-03	2.71e-04	-1.54e-03	-1.99e-04	-1.98e-04
20	-1.68e-04	3.58e-03	-1.11e-03	-1.45e-04	5.51e-04	3.85e-03	3.81e-03
21	1.01e-04	5.54e-03	-1.11e-03	-8.08e-05	-7.87e-04	7.53e-03	7.13e-03
22	3.45e-04	4.10e-03	-4.59e-04	-2.51e-05	-5.19e-04	8.52e-03	7.14e-03
23	1.40e-04	1.66e-03	-1.51e-04	-7.07e-06	-1.90e-04	5.66e-03	4.05e-03
24	4.17e-05	5.35e-04	-4.69e-05	-2.10e-06	-6.28e-05	2.32e-03	1.50e-03
25	1.28e-05	1.35e-04	-9.31e-06	-5.65e-07	-1.45e-05	6.39e-04	4.00e-04
zone	hco	co2	ch3	ch4	ho2	h2o2	ch2o
1	-1.33e-04	9.45e-06	-1.85e-04	7.26e-04	-2.86e-05	-1.44e-05	-1.24e-04
2	-4.56e-04	3.30e-05	1.47e-03	2.47e-03	-9.90e-05	-3.08e-05	-4.33e-04
3	-5.56e-04	5.00e-05	4.60e-03	4.69e-03	-1.19e-04	7.50e-05	-5.81e-04
4	2.17e-05	4.82e-05	7.52e-03	7.52e-03	2.04e-05	4.77e-04	6.44e-05
5	1.41e-03	3.76e-05	1.14e-02	1.14e-02	3.48e-04	1.27e-03	6.44e-03
6	5.87e-03	1.14e-05	1.68e-02	1.68e-02	8.33e-04	2.48e-03	1.60e-02
7	1.84e-02	-1.12e-04	2.42e-02	2.42e-02	1.29e-03	3.90e-03	2.43e-02
8	3.34e-02	-5.98e-04	3.40e-02	3.41e-02	1.27e-03	4.88e-03	3.42e-02
9	4.64e-02	-1.99e-03	4.64e-02	4.66e-02	7.19e-04	3.96e-03	4.66e-02
10	5.90e-02	-4.78e-03	6.01e-02	6.04e-02	8.83e-03	8.36e-04	5.97e-02
11	6.31e-02	-8.14e-03	6.83e-02	6.91e-02	3.95e-02	2.53e-02	6.69e-02
12	4.68e-02	-9.52e-03	6.04e-02	6.44e-02	6.02e-02	7.80e-02	6.11e-02
13	9.73e-03	-8.36e-03	3.77e-02	4.88e-02	4.50e-02	8.10e-02	4.16e-02
14	-2.88e-02	-6.98e-03	1.32e-02	3.23e-02	1.29e-02	3.68e-02	1.68e-02
15	-4.69e-02	-6.41e-03	-2.32e-03	2.01e-02	-7.62e-03	6.08e-03	3.35e-04
16	-4.17e-02	-6.09e-03	-6.80e-03	1.20e-02	-1.15e-02	-2.71e-03	-4.87e-03
17	-2.54e-02	-5.39e-03	-5.84e-03	6.46e-03	-8.88e-03	-4.34e-03	-5.15e-03
18	-9.74e-03	-3.52e-03	-3.56e-03	2.45e-03	-4.46e-03	-2.92e-03	-3.42e-03
19	-2.05e-03	-2.00e-04	-2.39e-04	3.08e-04	-6.29e-04	-4.05e-04	-2.33e-04
20	-1.30e-03	3.84e-03	2.91e-03	-1.55e-04	7.04e-04	8.92e-04	2.82e-03
21	-1.28e-03	7.27e-03	3.41e-03	-8.57e-05	3.92e-04	6.29e-04	3.25e-03
22	-5.61e-04	7.44e-03	1.76e-03	-2.64e-05	1.02e-04	2.08e-04	1.66e-03
23	-1.99e-04	4.32e-03	5.62e-04	-7.42e-06	2.27e-05	5.65e-05	5.33e-04
24	-6.86e-05	1.60e-03	1.64e-04	-2.19e-06	5.29e-06	1.53e-05	1.58e-04
25	-2.12e-05	4.23e-04	3.98e-05	-5.86e-07	1.33e-06	3.74e-06	3.89e-05

Table A.14: Edit of the relative changes in the physical variables due to the remap onto the moving grid of the methane test problem at cycle 1500.

remap relative change per cycle

zone	ch3o	t	dx	u
1	-1.58e-04	-5.21e-06	7.66e-04	1.00e+00
2	1.49e-03	-1.95e-05	7.58e-04	1.02e-03
3	4.59e-03	-3.38e-05	7.78e-04	-1.52e-05
4	7.51e-03	-4.43e-05	8.23e-04	-2.19e-02
5	1.14e-02	-6.55e-05	8.75e-04	6.58e-03
6	1.67e-02	-1.17e-04	9.12e-04	5.50e-03
7	2.39e-02	-2.14e-04	9.13e-04	5.75e-03
8	3.35e-02	-3.55e-04	8.78e-04	6.23e-03
9	4.60e-02	-5.21e-04	8.28e-04	6.68e-03
10	6.10e-02	-8.08e-04	6.42e-04	7.63e-03
11	7.09e-02	-2.02e-03	-6.54e-04	1.23e-02
12	5.69e-02	-5.17e-03	-4.42e-03	2.13e-02
13	1.43e-02	-8.67e-03	-9.08e-03	2.39e-02
14	-3.15e-02	-9.91e-03	-1.14e-02	1.80e-02
15	-5.08e-02	-8.91e-03	-1.11e-02	1.01e-02
16	-4.22e-02	-6.91e-03	-9.85e-03	3.94e-03
17	-2.01e-02	-4.50e-03	-8.30e-03	-3.12e-04
18	-2.24e-03	-1.99e-03	-6.49e-03	-3.09e-03
19	1.07e-03	-2.78e-04	-3.77e-03	-3.63e-03
20	-8.18e-04	1.42e-04	-2.89e-03	-3.23e-03
21	-5.25e-04	7.89e-05	-2.73e-03	-2.93e-03
22	-2.02e-04	2.48e-05	-2.79e-03	-2.82e-03
23	-7.15e-05	7.11e-06	-2.87e-03	-2.85e-03
24	-2.68e-05	2.19e-06	-2.92e-03	-2.90e-03
25	-1.25e-05	6.24e-07	-2.93e-03	-2.92e-03

Table A.14: Edit of the relative changes in the physical variables due to the remap onto the moving grid of the methane test problem at cycle 1500: continued.

remap relative change per cycle

zone	h	h2	o	o2	oh	h2o	co
1	1.12e-03	7.46e-04	8.11e-04	-8.80e-05	4.77e-04	-1.36e-04	8.54e-04
2	3.53e-03	2.38e-03	2.60e-03	-3.01e-04	1.58e-03	-4.56e-04	2.70e-03
3	3.51e-03	2.43e-03	2.73e-03	-4.52e-04	1.86e-03	-6.52e-04	2.64e-03
4	-1.78e-03	-9.84e-04	-7.40e-04	-4.50e-04	3.00e-04	-5.72e-04	-1.53e-03
5	-1.00e-02	-6.26e-03	-6.14e-03	4.59e-04	-2.11e-03	-4.42e-04	-8.46e-03
6	-1.70e-02	-1.06e-02	-1.06e-02	-6.10e-04	-3.83e-03	-4.57e-04	-1.67e-02
7	-1.94e-02	-1.18e-02	-1.19e-02	-9.01e-04	-3.84e-03	-6.56e-04	-2.61e-02
8	-1.46e-02	-8.43e-03	-8.41e-03	-1.25e-03	-1.67e-03	-1.00e-03	-3.39e-02
9	-2.63e-03	-9.24e-04	-3.21e-04	-1.53e-03	2.35e-03	-1.34e-03	-3.44e-02
10	1.08e-02	4.31e-03	9.19e-03	-1.72e-03	6.92e-03	-1.31e-03	-2.55e-02
11	1.41e-02	-9.59e-04	1.32e-02	-1.95e-03	9.33e-03	-6.00e-04	-1.22e-02
12	5.29e-03	-3.27e-03	5.68e-03	-9.02e-04	4.16e-03	-6.05e-05	-2.61e-03
13	-6.84e-03	2.80e-03	-1.26e-02	2.60e-03	-1.17e-02	-9.92e-04	-6.76e-04
14	-2.07e-02	1.52e-03	-3.39e-02	6.34e-03	-3.20e-02	-2.98e-03	-2.74e-03
15	-4.13e-02	-3.90e-03	-5.41e-02	9.06e-03	-5.24e-02	-5.53e-03	-5.50e-03
16	-7.59e-02	-9.29e-03	-7.10e-02	1.17e-02	-7.24e-02	-9.59e-03	-9.57e-03
17	-1.03e-01	-1.75e-02	-8.07e-02	1.44e-02	-8.81e-02	-1.77e-02	-1.76e-02
18	-1.02e-01	-3.31e-02	-7.63e-02	1.47e-02	-9.65e-02	-3.33e-02	-3.32e-02
19	-3.97e-02	-5.29e-02	-1.60e-02	9.66e-03	-8.22e-02	-5.36e-02	-5.34e-02
20	2.69e-03	-6.25e-02	2.11e-02	3.55e-03	-2.49e-02	-6.60e-02	-6.55e-02
21	-8.90e-05	-4.78e-02	1.06e-02	7.93e-04	6.87e-03	-6.21e-02	-5.93e-02
22	-1.74e-03	-2.07e-02	2.42e-03	1.34e-04	2.70e-03	-4.00e-02	-3.44e-02
23	-3.83e-04	-4.45e-03	4.07e-04	1.95e-05	5.08e-04	-1.45e-02	-1.06e-02
24	-3.52e-05	-4.88e-04	4.62e-05	1.91e-06	5.91e-05	-2.04e-03	-1.34e-03
25	1.33e-06	1.43e-05	-1.02e-06	-8.16e-08	-1.57e-06	6.77e-05	4.24e-05
zone	hco	co2	ch3	ch4	ho2	h2o2	ch2o
1	1.82e-03	-1.29e-04	2.54e-03	-9.77e-03	3.90e-04	1.97e-04	1.69e-03
2	5.97e-03	-4.31e-04	-1.76e-02	-3.21e-02	1.29e-03	4.14e-04	5.67e-03
3	7.02e-03	-6.15e-04	-5.50e-02	-5.64e-02	1.50e-03	-7.75e-04	7.25e-03
4	6.36e-04	-5.34e-04	-7.92e-02	-7.92e-02	-1.79e-05	-4.68e-03	4.30e-04
5	-1.12e-02	-3.55e-04	-1.00e-01	-1.01e-01	-2.80e-03	-1.07e-02	-4.62e-02
6	-3.64e-02	-1.20e-04	-1.18e-01	-1.18e-01	-5.65e-03	-1.70e-02	-1.11e-01
7	-8.88e-02	4.72e-04	-1.30e-01	-1.31e-01	-7.05e-03	-2.10e-02	-1.31e-01
8	-1.31e-01	1.99e-03	-1.34e-01	-1.34e-01	-5.57e-03	-1.99e-02	-1.35e-01
9	-1.25e-01	4.65e-03	-1.25e-01	-1.26e-01	-2.42e-03	-1.25e-02	-1.26e-01
10	-9.97e-02	7.12e-03	-1.01e-01	-1.01e-01	-9.33e-03	-2.96e-03	-1.01e-01
11	-5.66e-02	6.47e-03	-5.97e-02	-6.02e-02	-2.41e-02	-9.83e-03	-5.87e-02
12	-1.49e-02	2.28e-03	-1.65e-02	-1.67e-02	-1.35e-02	-9.72e-03	-1.60e-02
13	-1.36e-03	-1.61e-03	5.91e-03	8.93e-03	7.20e-03	1.43e-02	6.88e-03
14	-1.70e-02	-3.63e-03	5.69e-03	1.62e-02	4.90e-03	1.65e-02	7.44e-03
15	-4.26e-02	-5.74e-03	-3.08e-03	1.71e-02	-7.83e-03	3.98e-03	-6.70e-04
16	-6.36e-02	-9.67e-03	-1.09e-02	1.77e-02	-1.82e-02	-5.24e-03	-8.16e-03
17	-7.40e-02	-1.77e-02	-1.88e-02	1.87e-02	-2.75e-02	-1.47e-02	-1.71e-02
18	-6.95e-02	-3.35e-02	-3.34e-02	1.73e-02	-3.66e-02	-2.67e-02	-3.24e-02
19	-9.85e-03	-5.38e-02	-5.15e-02	1.07e-02	-3.39e-02	-3.16e-02	-5.05e-02
20	2.51e-02	-6.61e-02	-5.34e-02	3.82e-03	-1.68e-02	-1.98e-02	-5.19e-02
21	1.22e-02	-6.03e-02	-3.10e-02	8.41e-04	-3.89e-03	-6.10e-03	-2.96e-02
22	2.93e-03	-3.56e-02	-9.22e-03	1.41e-04	-5.54e-04	-1.11e-03	-8.68e-03
23	5.38e-04	-1.13e-02	-1.53e-03	2.04e-05	-6.40e-05	-1.56e-04	-1.45e-03
24	5.78e-05	-1.44e-03	-1.52e-04	2.00e-06	-4.81e-06	-1.42e-05	-1.46e-04
25	-2.27e-06	4.49e-05	4.21e-06	-8.38e-08	1.20e-07	3.76e-07	4.11e-06

Table A.15: Edit of the relative changes in the physical variables due to the remap onto the cycle 1500 grid of the methane test problem at cycle 1501.

zone	ch3o	t	dx	u
1	2.17e-03	7.10e-05	-1.07e-02	1.00e+00
2	-1.78e-02	2.54e-04	-9.35e-03	-1.38e-02
3	-5.49e-02	4.12e-04	-6.83e-03	1.78e-04
4	-7.91e-02	4.73e-04	-3.24e-03	1.44e-01
5	-1.00e-01	5.70e-04	1.34e-03	-6.12e-02
6	-1.17e-01	8.00e-04	6.57e-03	-3.39e-02
7	-1.29e-01	1.12e-03	1.20e-02	-2.16e-02
8	-1.32e-01	1.37e-03	1.73e-02	-1.11e-02
9	-1.24e-01	1.40e-03	2.23e-02	-6.20e-04
10	-1.02e-01	1.28e-03	2.68e-02	8.90e-03
11	-6.16e-02	1.32e-03	2.99e-02	1.46e-02
12	-1.75e-02	6.15e-04	2.78e-02	1.89e-02
13	-1.07e-03	-2.02e-03	1.76e-02	2.34e-02
14	-1.88e-02	-5.19e-03	9.51e-03	2.33e-02
15	-4.59e-02	-7.80e-03	9.24e-03	2.37e-02
16	-6.35e-02	-1.05e-02	1.12e-02	2.68e-02
17	-5.06e-02	-1.34e-02	1.39e-02	3.12e-02
18	1.85e-02	-1.42e-02	1.42e-02	3.27e-02
19	4.43e-02	-9.61e-03	1.04e-02	2.59e-02
20	1.94e-02	-3.50e-03	5.08e-03	1.43e-02
21	5.08e-03	-7.74e-04	-2.81e-04	4.26e-03
22	1.05e-03	-1.32e-04	-5.21e-03	-2.54e-03
23	1.97e-04	-1.94e-05	-9.12e-03	-7.25e-03
24	1.71e-05	-1.92e-06	-1.17e-02	-1.05e-02
25	-1.34e-06	6.65e-08	-1.24e-02	-1.21e-02

Table A.15: Edit of the relative changes in the physical variables due to the remap onto the cycle 1500 grid of the methane test problem at cycle 1501: continued.

Appendix B

Sample NAMELIST Input for Various Types of Problems

B.1 CH4-72

```
mboxid(1)=8hbox t55
mboxid(2)=16h72 ch4
    lists=1:6 8 12:20 24:26 29 30 33 34 100 23
    lists(26)=22
lists(27)=113 160
lists(29)=141 273 274
lists(32)=168 169 174 175 180
listr(1)=1552 2 172 4 62 6 7 708 9:12 75 73
listr(15)= 365 56 274 82 67 605 69 70 705
    listr(24)=61 25:40 150 42:54 13 14 15
    listr(58)=104:106 363 364 109 93 94 90 91 95 92 96:103
    listr(78)=57:60 78:80
    listr(85)=277 74 152 145
listr(89)=483 207 428 429 208 489 491:496
listr(101)=610 665 482 694 695 699:702
listr(110)=151 719 721 148 1161 66
listr(116)=573 1111:1129 1129 410
listr(138)=615 616 653 654 666 641 643 1049 1050
listr(147)=1692 1693 1663 1664 1015 718 1595
listr(154)=659 1377 660 1374 1375
nzones=1
xmax=1.0e-4
iprint=6(20)
iprint(2)=0
iprint(3)=2
iprint(1)=10
iprint(5)=10
lste4=1
    dzero=1.92e-6
    dczero=1.75e-6
    qk=.2
    fnertia=1.e-8
    eta2=.2
    delymn=20.
    euler=1.
    euler2=-1.
    rzflag=0.
```

```

      itermx=20
      rdelcmx=.1
dt=1.0e-11
      dtmin=1.e-20
      maxdump=300
      frc(22)=1.6e13
      rrc(22)=1.0e10
      frc(37)=4.8e13
      rrc(37)=3.04e14
      frc(65)=4.0e12
      frx(65)=2.8e4
      rrc(65)=1.0e11
      rrx(65)=6.365e4
edit4mn=1.0e-4
      frc(85)=1.0e12
      rrc(85)=1.0e12
      frc(86)=2.0e12
      rrc(86)=6.0e11
      frc(88)=2.0e11
      rrc(88)=0.001
      frc(62)=4.8e12
      rrc(62)=1.2e12
      rrx(62)=1.4e4
      frcmul(46)=1.2
      rrcmul(46)=1.2
      frcmul(22)=2.0
      rrcmul(22)=2.0
listre=25 26 57 51
      eff(6)=20.0
      eff(37)=1.0 3.3 1.0 1.0 1.0 21.0 1.0 2.0 1.0 5.0
      eff(73)=2.0 3.0 1.0 1.0 1.0 6.0
eff(109)=3(1.0) 0.78 1.0 7.0 7(1.0) 7.7
frc(87)=1.0e-20
rrc(87)=1.0e-20
frc(44)=7.6e10
frx(44)=2.7e3
* use pressure dependent reactions:
listrf(1)=1 15 24 60
arc(1)=6.3e-3
arx(1)=18.0e3
listr(15)=1550
arc(2)=3.47e-12
arp(2)=2.76
arx(2)=2120.0

```

```

listr(24)=1551
arc(3)=7.66e-7
arx(3)=3032.0
listr(60)=1553
arc(4)=3.72e-9
arp(4)=1.12
* warnatz's ho2+ho2:
listr(50)=1555
* use Baldwin's ho2+ho2 (jcs trans 2, 82, 1445 (1986)):
rrc(50)=3.02e12
rrx(50)=1.39e3
* correct r+o2=ro2 rates:
frc(138)=4.39e18
frp(138)=-1.0
frx(138)=3.702e4
* use new ch4+ho2:
listr(53)=1704
* use new c2h4+oh:
listr(62)=1656
* correct ch3+ro2h=ch4+ho2 to give better ch4 concentrations in 563K case:
*; rrx(53)=1.28e3
* reduce ch3+rooh to reduce ch4 and increase ch3o2 reactions:
rrcmul(53)=0.2
rrcmul(147)=2(0.2)
* 3-5-87
* use falloff from Tsang for ch3o2=ch3+o2:
frcmul(139)=0.1
rrcmul(139)=0.1
* fix reverse of pc2h4oh and sc2h4oh:
rrc(135)=2(1.0e11)
rrp(135)=2(0.0)
rrx(135)=2(0.0)
* assume all c2h4oh is pc2h4oh (so as to avoid ch3cho production):
rnlhs(1,136)=32hsc2h4oh
irs(136,1)=274
* use miller's rate for ch2+ch3=c2h4+h:
frc(88)=3.0e13
* add c2h6+ch3o2,c2h5o2:
listr(159)=1702 1781
* fix reverse rate:
rrc(159)=2(7.5e11)
rrx(159)=2(1.28e3)
* use walker's rate for c2h6+o2 (4x faster):
frc(101)=4.0e13

```

```

frx(101)=50.9e3
rrx(147)=6.72e3
rrcmul(53)=1.0
rrcmul(147)=1.0
* use baldwins, jones and walker's ch4+ho2:
frc(53)=1.12e13
frx(53)=24.64e3
rrc(53)=7.43e11
rrx(53)=5.50e3
* ch4+ch3o2 by analogy:
frc(147)=1.12e13
frx(147)=24.64e3
rrc(147)=7.43e11
rrx(147)=5.5e3
* tsang's rate for c2h4+ch3
frc(137)=6.62
frp(137)=3.7
frx(137)=9.5e3
rrc(137)=1.44
rrp(137)=4.02
rrx(137)=5.472e3
* Tsang's rate for ch4+ch3o:
frc(108)=1.57e11
frx(108)=8.842e3
* add ch3+o2=ch2o+oh, Saito et al.
listr(161)=143
frc(161)=3.16e11
frx(161)=9.0e3
rrc(161)=3.29e11
rrx(161)=6.251e4
* correct error in cdat:
rrx(5)=1.256e4
* use clark and dove expression for ch4+h and Tsang's reverse rate
* forward rate is the same at 1100 K and Tsang's reverse rate is 35% slower
listr(2)=283
rrc(2)=289.0
rrp(2)=3.12
rrx(2)=8.711e3
* use cohen and westenberg's ch4+oh (about 2x slower at 1100 K)
listr(3)=3586
* use Tsang's rate for ch3o decomposition ( about 21 times faster )
frc(40)=3.888e37
frp(40)=-6.65
frx(40)=33.26e3

```

```

keq(40)=1
* zero ch3+o2=ch2o+oh
frc(161)=0.0
rrc(161)=0.0
*
dtmax=2.0
mxcycle=3000
maxdump=600
timemx=30.0
pr0=1.013e6
c=36(1.0e-15) 1000. 1.0e-4 0.0
c(12)=1.0
c(4)=2.0
c(7)=7.52
endin=1.0
$
$

```

B.2 SANT89

The following problem runs to cycle 677 time=2.5e-1 dt=1.998e-18 and temperature in zone 1=917.

```

mboxid='box t55 89 6 atm 898k'
  lists=1:6 8 12:20 24:26 29 30 33 34 100 23
  lists(26)=22
  lists(27)=109 110 114 115
lists(31)=142 143
lists(33)=121 166 145
lists(36)=113 160
lists(38)=27 147 148 146 144 118
lists(44)=151 362 150 167 214 218 141 273 274
lists(53)=153 371
lists(55)=168 169 170 171 215 192 226 373
lists(63)=174:177 216 190 228 374
lists(71)=180:182 217 191 227 375 406
lists(79)=346 347 387:390
lists(85)=149 372
listr(1)=1552 2 172 4 62 6 7 708 9:12 75 73
listr(15)= 365 56 274 82 67 605 69 70 705
  listr(24)=61 25:40 150 42:54 13 14 15

```



```

      listr(58)=104:106 363 364 109 93 94 90 91 95 92 96:103
      listr(78)=57:60 78:80
      listr(85)=277 74 152 145
      listr(89)=228:232 238:242
      listr(99)=284:287 291:296
      listr(109)=302:304 368 369 406 407
      listr(116)=503 502 415 416 422 414 425 432 433 417
      listr(126)=288 289 404 405 300 301 580 581 587:592 583:586 603
      listr(145)=483 207 428 429 208 489 491:496
      listr(157)=607:609
      listr(160)=402 403 246
      listr(163)=610 665
      listr(165)=395 397 399 393
      listr(169)=396 398 400 394
      nzones=1
      xmax=1.0e-4
      iprint=6(20)
      iprint(2)=0
      iprint(3)=3
      lste4=1
      dzero=1.92e-6
      dczero=1.75e-6
      reldc(1)= 5.29 3.74 1.32 0.94 1.28 1.25 1.00 1.00 0.98 0.80
      reldc(11)=1.37 1.32 0.92 0.91 0.97 0.95 0.97 1.00 0.98 1.41
      reldc(21)=1.47 1.06 1.04 1.02 0.94
      reldc(26)=0.95
      reldc(27)=0.80 0.82 0.81 0.81
      reldc(31)=0.81 0.80
      reldc(33)=0.83 0.84 0.71
      reldc(36)=0.82 0.83
      reldc(38)=0.69 0.70 0.70 0.71
      reldc(42)=0.71 0.72
      reldc(44)=0.63
      qk=.2
      fnertia=1.e-8
      eta2=.2
      delymn=20.
      euler=1.
      euler2=-1.
      rzflag=0.
      itermx=20
      rdelcmx=.1
      dt=1.0e-11
      dtmin=1.e-20

```

```

maxdump=600
frc(22)=3.2e13
rrc(22)=2.0e10
frc(37)=4.8e13
rrc(37)=3.04e14
frc(65)=4.0e12
frx(65)=2.8e4
rrc(65)=1.0e11
rrx(65)=6.365e4
edit4mn=5.0e-4
frc(85)=1.0e12
rrc(85)=1.0e12
frc(86)=2.0e12
rrc(86)=6.0e11
listre=25 26 57 51
eff(6)=20.0
eff(94)=1.0 3.3 1.0 1.0 1.0 21.0 1.0 2.0 1.0 5.0
eff(187)=2.0 3.0 1.0 1.0 1.0 6.0
eff(280)=3(1.0) 0.78 1.0 7.0 7(1.0) 7.7
frc(87)=0.0
rrc(87)=0.0
listr(173)=482
rrc(169)=1.0e12
listr(174)=672:678 680:689 694:696 698:702
listr(199)=151
listr(200)=706 707
frc(44)=7.6e10
frx(44)=2.7e3
listr(202)=719:721 148
frc(143)=3.e13
frc(143)=1.e12
frcmul(143)=2.
frcmul(143)=6.
frcmul(143)=30.
* use pressure dependent reactions:
listrf(1)=1 15 24 60
arc(1)=6.3e-3
arx(1)=18.0e3
listr(15)=1550
arc(2)=3.47e-12
arp(2)=2.76
arx(2)=2120.0
listr(24)=1551
arc(3)=7.66e-7

```

```

arx(3)=3032.0
listr(60)=1553
arc(4)=3.72e-9
arp(4)=1.12
* use Baldwin's ho2+ho2 (jcs trans 2, 82, 1445 (1986)):
*(factor of 2 slower than Walker, 1975 and Warnatz, 1984. at 563k)
listr(50)=1649
* use tully's modified rate for c3h8+oh:
listr(105)=1586 1587
listr(206)=1161 66 4376 1571 1572
listr(211)=1588 1590 1581:1585 1591 1589
listr(220)=573 1111:1126 1129:1131
listr(240)=1080 1081 801 803 802 805 804 809
listr(248)=1083 1084 1087 1088 1090 1091 806 810
listr(256)=807 811 1091 1092 1082 808
listr(262)=766:774 776:781 1681 1682 1686:1688
listr(282)=1013 412 1064 1065 1068 1166 1063 1066 1067 1639 1640
listr(293)=1168 1169
listr(295)=611 612 615 616 712 1095 1629 761
listr(303)=653:656 715 1097 1630 763
listr(311)=666 667 787 668 764 1626 765 714 1098 1631
listr(321)=709 794
listr(323)=629 630 625 626 621 622 617 618
listr(331)=799 800 1101 1102 1633 1634 1074 1075
listr(339)=641 643 645 647 1676 1110 1677
listr(346)=642 644 646 648 711 1109 1678 1165
listr(354)=1049:1052 1060 1105 1691
listr(361)=1043:1046 1059 1106 1632 762
listr(367)=1078 1061 1107 1079 1062 1108
listr(373)=1168 1169
listr(375)=1692:1695 1700 1099 1701
listr(382)=1663:1669
listr(389)=1015 1018 1019 1037 1752
listr(394)=718 1595:1597 1455 1753
listr(400)=659 1377 660:662 812 713 1372 1096
listr(409)=1374 1375 1370 1372 1373
listr(414)=386 388 390 389 606 387 391 392
listr(422)=956 410 793 1376
* use new ch4+ho2:
listr(53)=1704
* use new c2h4+oh:
listr(62)=1656
* add isomerization reactions:
lists(87)=404

```

```

listr(426)=1727 1728
listr(428)=1746:1751
listr(434)=1754 1757 1758
listr(437)=431
* 3 march 87
* decrease ch3co=ch3+co by 5.5 to account for falloff (Tsang):
* at 950 K
frcmul(125)=0.18
rrcmul(125)=0.18
* correct ch3+ro2h=ch4+ho2 to give better ch4 concentrations in 563K case
rrx(53)=1.28e3
listr(439)=602
* correct reverse rate of c2h5co2:
rrc(318)=2(1.0e10)
rrx(318)=2(5.0e4)
* reduce ch3+rooh to reduce ch4 and increase ch3o2 reactions:
rrcmul(53)=0.2
rrcmul(375)=9(0.2)
* 3-5-87
* use falloff from Tsang for ch3o2=ch3+o2:
frcmul(298)=0.1
rrcmul(298)=0.1
* 3-5-87
* use baker, baldwin, fuller and walker's, c4h9+o2 rates at 753 K:
* based on allara and shaws decomposition rates for pc4h9 and sc4h9 and
* baker et al.'s yields.
* add ro2+c4h8 reactions:
listr(441)=1761:1765 1768:1773 1776 1779:1782
* use new rate for h2+o:
listr(9)=1783
* add hcco+o2 reaction:
listr(438)=1365
* add c3h6+o2 reaction:
listr(457)=1038
* miller's rate for ch2+ch3=c2h4+h:
frc(88)=3.0e13
rrc(88)=0.0
* use Miller's current hcco rates:
listr(154)=1364
listr(156)=1366
* put in Miller's ch3+oh=ch2+h2o and ch3+oh=ch2oh+h for future use:
listr(459)=4483
frc(459)=0.0
* add to Ea of c4h8+o2 to reduce rate, since it is faster than c4h8+oh:

```

```

frx(453)=3.7e4
frx(454)=4.0e4
* chemistry input file from Run 1725, green, nbutane
* 16 Sept 86
* decrease ch3co=ch3+co to account for falloff (Tsang):(at Wilk's conditions)
frcmul(125)=0.48
rrcmul(125)=0.48
* reduce irreversible reaction of c4h8+oh by addition channel:
frc(138)=2(1.0e11)
frc(166)=1.0e11
frc(170)=1.0e11
* reduce ch3+o2=ch3o2 to agree with Tsang's falloff at Wilk's conditions:
frcmul(298)=0.5
rrcmul(298)=0.5
* add other path for c2h5o and c3h7o decomposition:
listr(460)=788 786
* use Tully's new 1c4h8+oh rate:
* add waddington mechanism for c3h6:
listr(462)=1544:1546
* reduce irreversible addition of oh to propene:
frc(114)=1.0e11
frc(116)=1.0e11
* use Tsangs rate for h2o2 decompostion at Wilk's conditions:
frcmul(51)=0.5
rrcmul(51)=0.5
* reduce ch4 concentrations:
* (used rrcmul=0.05 for propene)
rrcmul(53)=0.08
rrcmul(375)=9(0.08)
* reduce ch4 concentrations:
* (used rrcmul=0.05 for propene)
rrcmul(53)=0.056
rrcmul(375)=9(0.056)
* reduce ch4 concentrations:
* (used rrcmul=0.05 for propene)
rrcmul(53)=0.001
rrcmul(375)=9(0.001)
* use Allara and Shaw's new rate for ch3+c4h10:
* to estimate rate for ch3+c3h8
* (were using allara and edelson's old #'s)
frc(90)=1.99e11 3.98e11
frx(90)=9.6e3 11.4e3
* add some disporportionation reactions:
* involving c4h7 and alkyl radicals

```

```

listr(465)=419:421
* add h2o2+ch3o2 = ho2+ch3o2h
listr(468)=3554 3553
* ro2+r'o2=ro+r'o+o2 x .5:
frcmul(410)=2(0.5)
frcmul(413)=0.5
frcmul(456)=0.5
frcmul(400)=5(2.0)
rrcmul(400)=5(2.0)
* include ro2+ho2=ro+oh+o2 path:
listr(470)=660:662
* speed up c4h9+o2=c4h9o2 both ways at 563k and add positive Ea:
rrc(295)=3(3.4e11)
rrx(295)=3(-2.0e3)
frc(295)=2.53e17
frx(295)=33.2e3
frc(297)=2.53e17
frx(297)=33.2e3
frc(296)=1.49e18
frx(296)=35.02e3
* slagle and gutman's ch3+o2=ch3o2:
listr(298)=4474
* use Tsangs rate at 715k for ch3o decomp:
frc(40)=3.888e37
frp(40)=-6.65
frx(40)=3.326e4
rrc(40)=7.706e32
rrp(40)=-6.65
rrx(40)=9.697e3
* from Baker et al. runs:
* reduce ho2+ch3cho by .71 to agree with Kaiser's (Baldwin's)
* c2h5cho+ho2 and lower rate of ch3cho consumption in Baker's
* runs and in 715k case:
frc(122)=1.20e12
* add reactions added to cdat v. 131:
listr(473)=3551 3552 3555 3556
* replace reactions put in manually with ones in cdat:
listr(440)=3594
listr(458)=3588
* zero reactions thought not to occur:
frc(95)=0.0
rrc(95)=0.0
* (check with Mel Colkett about c2h2+o2)
frc(85)=0.0

```

```

rrc(85)=0.0
frc(133)=0.0
rrc(133)=0.0
* 11/1/88:
* add propyl peroxy isomerization reactions:
lists(88)=950:955
listr(477)=4457:4471
* account for falloff of  $\text{CH}_3+\text{O}_2$  at 10atm and 900K
*(use Tsang's results)
frcmul(298)=0.106
rrcmul(298)=0.106
* use tsang's rate for propyl decomposition:
frc(96)=1.26e13
frx(96)=30.4e3
frc(94)=2.19e13
frx(94)=37.16e3
* use Walker's 22nd Combustion Symp.  $\text{CH}_2\text{O}+\text{HO}_2$ :
frc(54)=5.6e12
frx(54)=13.6e3
keq(54)=1
* use gutman's new  $\text{CH}_3\text{O}+\text{O}_2$  (Timonen, Ratajczak, gutman, J. Phys. Chem, in press)
frc(46)=4.2e12
keq(46)=1
* use tsangs  $\text{HCO}+\text{M}$ :
frc(12)=5.12e21
frp(12)=-2.14
frx(12)=20.42e3
keq(12)=1
* zero direct reaction,  $\text{C}_3\text{H}_7+\text{O}_2=\text{C}_3\text{H}_6+\text{HO}_2$ :
frc(110)=2(0.0)
rrc(110)=2(0.0)
* zero  $\text{C}_3\text{H}_6+\text{HO}_2=\text{C}_3\text{H}_6\text{OOH}$  direct reaction:
frc(282)=0.0
rrc(282)=0.0
* reduce  $\text{C}_3\text{H}_6+\text{HO}_2=\text{C}_3\text{H}_6\text{OOH}$ :
rrx(483)=3(8.0e3)
* reduce  $\text{C}_3\text{H}_7\text{O}_2=\text{C}_3\text{H}_6\text{OOH}$  to reduce  $\text{C}_3\text{H}_6\text{O}$  and oxidation rate:
frcmul(477)=3(0.2)
* correct  $\text{C}_2\text{H}_4+\text{RO}_2$ :
frc(354)=7(1.13e13)
frx(354)=7(30.43e3)
* correct  $\text{C}_2\text{H}_3+\text{O}_2$ :
listr(206)=1420
* zero duplicate reactions:

```

```

frc(471)=2(0.0)
rrc(471)=2(0.0)
frc(403)=2(1.0e12)
* use Tsangs new ch3+c3h8 ( about 15% faster, and gives more isopropyl):
frc(90)=1.506 0.903
frp(90)=3.46 3.65
frx(90)=5.48e3 7.135e3
rrc(90)=7.23e-4 2.4e-2
rrp(90)=4.4 4.02
rrx(90)=10.8e3 10.87e3
* use Tsangs ch3+ho2=ch4+o2 (4 times faster). Will increase ch4
* and should reduce oxidation rate:
frc(45)=3.61e12
frx(45)=0.0
rrc(45)=4.0e13
rrx(45)=56.9e3
* use Tsang's ch2o+ch3 (4 times faster). Will increase ch4,
* and should reduce oxidation rate:
frc(38)=5540.
frp(38)=2.81
frx(38)=5.86e3
rrc(38)=7290.
rrp(38)=2.85
rrx(38)=22.51e3
* ch3o2+ch3o2 set old estimate. Use Anastasi et al., Gas Kinetics,
* ch3o2+ch3o2=ch2o+ch3oh+o2 set to old Kaiser estimate. Use Anastasi,
* Couzens, Waddington et al.'s new values. Faraday Trans(?) reprint.
* Also given at Swansea, 1988. Should help increase ch4:
* Swansea, value which agrees well with Kaiser. Should increase ch4:
* should slow rate down too.
frc(409)=2.8e9
frx(409)=-2.18e3
frc(410)=5.7e11
frx(410)=1.84e3
* Reevaluated ch3o2+ho2. Used Kurylo,Dagaut et al. for total rate at 298K
* Used Tsang for Ea. Split two paths equally:
frc(400)=2(1.1e10)
frx(400)=-2600.0
* reduce c3h6o2h=c3h6o+oh (x .2) Incorporate later into Ea which is low
* compared to Baldwin,Dean and Walker.
frcmul(480)=3(0.2)
* increase c3h7o2=c3h6o2h x 3 to increase c3h6 and decrease c2h4:
frcmul(477)=3(1.6)
rrcmul(477)=3(1.6)

```



```

* use Tsang's c3h8+ho2 rates which have curvature
* (changed frc to agree with Walker at 753K)
frc(107)=14750.0 5.52e4
frp(107)=2.6 2.55
frx(107)=13.91e3 16.48e3
rrc(107)=442.0
rrp(107)=2.83
rrx(107)=4.05e3
rrc(108)=2.17e4
rrp(108)=2.21
rrx(108)=2.57e3
* add c2h6+ro2:
listr(492)=1702 1781
* decrease c3h6o2h+o2 to decrease oxidation rate at high pressures:
frcmul(486)=3(0.3)
rrcmul(486)=3(0.3)
* assume 1-c3h6ooh-3 goes to c3h6o as in previous runs
* (change in future runs)
irs(481,2)=149
rnrhs(1,481)='c3h6o+oh'
*
* Litzinger and Santavicca
* propane, high pressure flow reactor
mxcycle=800
dtmax=0.05
* 4/20/88 initial conditions
timemx=0.25
pr0=6.078e6
prbr=6.078e6
teng=909.6 907. 902. 898. 895. 893. 892. 893. 897. 908. 917.
timdat=0. 6.61e-3 19.14e-3 32.73e-3 45.85e-3 58.55e-3
70.95e-3 83.14e-3 95.35e-3 107.42e-3 117.0e-3
wall=1.0e10
c=93(1.0e-15) 909.6 1.0e-4 0.0
c(27)=1.02e-2
c(4)=0.20794
c(7)=0.78186
endin=1.0
$
$

```

Appendix C

Definition of NAMELIST Variables

The following is a list of the variables that can be set in the input deck. See Table A.8 for an example of the input for a typical problem. It is recommended that one understand Table A.8 before attempting to generate a problem from scratch. A “0” is the number zero. “df” denotes the default value used by the code unless another value is specified in the input deck. We adhere to the FORTRAN convention that variables whose name begins with I through N are entered as integers. The defaults are given so that one can see whether a number is entered as an integer or a decimal number.

Descriptions of new variables were provided by Charles Westbrook and Bill Pitz.

ALAM0 Initial guess at λ in Eq. (6.7). df = 1.0. Usually not necessary to set.

ALPHAD The exponent of the temperature dependence of α^T . α^T is allowed the same form as D_i in Eq. (2.7). df = 0.5.

ALPHADC The exponent of the temperature dependence of D_i in Eq. (2.7). df = 0.5.

ARC(MXKF), ARP(MXKF), ARX(MXKF) See LISTRF.

MXKF (maximum number of reactions with falloff parameters) is 10.

ATWT(I) I = 1,IMX Atomic weight of species I. Is usually read from disk file CDAT, but can be modified in input deck.

BETAD The exponent of the dependence of α^T on c_{tot} . α^T is allowed the same form as D_i in Eq. (2.7). df = -1.0.

BETADC The exponent of the dependence of D_i on c_{tot} in Eq. (2.7). df = -1.0.

C(I,J) I = 1,IU; J = 1,JMX Allows specification of the physical variables $c_{i,j}$, T_j , Δr_j and $v_{j-\frac{1}{2}}$ at generation time. The array is read as a one-dimensional array with the variables for zone 1 followed by the variables for zone 2, and so forth. If C(I,J) is specified only for the

first zone, the same values will be assigned to all zones. This is the form used to initialize $C(I,J)$ in Table A.8. If all the variables are to be specified independently, remember that the dummy boundary zones must also be specified. This means that there are two more zones than the number of physical zones.

C0(I) I = 1,IT With $C1(I)$, $XDISC$, $XSPRD$ (and usually $PR0$) allows one to initialize $c_{i,j}$ and T_j with a step-function distribution $\alpha(x)$. Let (see Eq. (9.3))

$$\alpha(x) = \frac{1}{1 + \exp\left(\frac{x - XDISC}{XSPRD}\right)}$$

If $C0(IT)$ is nonzero, $C(I,J)$ will be initialized with a spatial dependence given by

$$C(I, J) = \alpha(x_j)C0(I) + [1 - \alpha(x_j)] C1(I) + \beta(x_j)C2(I).$$

For a description of $\beta(x)$, see $C2(I)$. Both species and temperature are set. The dynamic rezoner will choose Δr_j so that the step is described adequately, so $NZXCESS$ and $XSPRD$ should be nonzero.

C1(I) I = 1,IT See $C0$.

C2(I) I = 1,IT $C2(I)$, with $GAUSX$ and $GAUSDX$, allows one to specify an initial species and temperature distribution characterized by the function

$$\beta(x) = \frac{4}{[1 + \exp(-y)][1 + \exp(y)]}$$

where

$$y = \frac{x - GAUSX}{GAUSDX}$$

$\beta(x)$ has a maximum of 1 at $GAUSX$, and falls exponentially to zero over a distance characterized by $GAUSDX$. We use this form rather than a true Gaussian because it is the derivative of Eq. (9.3). Thus the indefinite integral analogous to Eq. (9.4) is easily done. See $C0$.

C1TVR(MXTVR), C2TVR(MXTVR), C3TVR(MXTVR) See $LISTTVR$.

$MXTVR$ (maximum number of time-varying rates) is 150.

CBURNR(MXIU) For problems with prescribed inlet boundary fluxes, this array contains the species concentrations, temperature, and boundary fluid velocity. Concentrations are renormalized to the boundary pressure.

CMPRATIO Compression ratio for free-piston (time-varying piston velocity) option. $df = 24.0$. See also IFPISTON.

CNEGF Species whose mole fractions are less than CNEGF are allowed to change RDELCNEG per cycle. $df = 1.E-6$.

CNEGFR Species whose mole fraction are less than CNEGFR are not advected from zone during remap. Usually not set. $df = 0.0$.

CNVG Convergence criteria. Code iterates until relative change per iteration in physical variable is less than CNVG. May be worth trying different values in very expensive calculations. $df = 1.E-6$.

CNVGRAT f of Eq. (4.17). $df = 0.25$.

COMPRESS Compression ratio. See RPM.

CPA1(I) I = 1,IMX Coefficient of $(T/1000)^0$ in expansion of c_p^i (see Eq. (2.15)). Units are *cal/mole*. Usually read from CDAT but can be modified in input deck. Modified value must be nonzero, since code takes CDAT value of CPA1 if not changed from zero by input deck. If real zero is desired, put in very small number. Index I must refer to index in problem, not index in CDAT.

CPA2(I) I = 1,IMX Coefficient of $(T/1000)^1$ in c_p^i . See CPA1.

CPA3(I) I = 1,IMX Coefficient of $(T/1000)^2$ in c_p^i . See CPA1.

CPA4(I) I = 1,IMX Coefficient of $(T/1000)^3$ in c_p^i . See CPA1.

CPA5(I) I = 1,IMX Coefficient of $(T/1000)^4$ in c_p^i . See CPA1.

CPA6(I) I = 1,IMX Coefficient of $(T/1000)^5$ in c_p^i . See CPA1.

CPCON If a fixed, constant specific heat is wanted, set CPCON equal to that constant value.

DADC If the dynamic rezoner is used, each cycle the code will calculate the value of the α of Eq. (6.2) needed to satisfy Eq. (6.5). DADC is the fraction of this correction to the current α that is made each cycle. Not usually modified. $df = 0.1$.

DCZERO The D_i^0 of Eq. (2.7) is taken as $DCZERO * RELDC(I)$. Default gives a number on the order of transport value. $df = 1.6E-6$.

DELYMN If $\int dx |dT/dx| < DELYMN$ dynamic rezoner will rezone problem with constant zoning. Set nonzero if dynamic rezoning desired. $df = 0.0$.

DELX(J) J = 1,JMX If dynamic rezoning is to be done, initial guess at Δr_j can be set with $DELX(J)$. Usually not set. $df = 0.0$.

DMUL is a multiplier for temperature conduction terms in DIFFUN and DIFPED. This variable was introduced when the Sandia transport package was added. $df = 1$.

DT Initial timestep in seconds. $df = 1.E-8$.

DTENG is the time increment when the variation of pressure (at the right boundary) or TWALLX is specified at constant time increments. For example, if pressure and/or temperature data are available from an engine at each crank angle, DTENG is the time corresponding to 1 crank angle degree.

DTMAX Maximum timestep in seconds to be allowed when problem is running. Must be set nonzero in input deck. $df = 0.0$.

DTMIN Minimum timestep in seconds to be allowed when problem is running. Pick value such that if problem drops to this timestep, it should stop before wasting more computer time. $df = 1.E-8$.

DTXMX Maximum timestep for next cycle is $DTXMX * DT$, where DT is current timestep. $df = 1.1$.

DZERO For α^T corresponds to D_i^0 of Eq. (2.7). See also DCZERO. $df = 1.6E-6$.

EDIT4MN Reaction rates less than $EDIT4MN$ times (sum of absolute values of rates affecting species) are not printed out in zone rate edit. See IPRINT(4). One can get all rates printed out by setting $EDIT4MN = 0.0$, $EDIT4MX = 1.0$. $df = 0.01$.

EDIT4MX When sum of reaction rates is at least EDIT4MX times (sum of absolute values of rates affecting species) no more reactions are printed out in zone rate edit. See IPRINT(4). One can get all rates printed out by setting EDIT4MN = 0.0, EDIT4MX = 1.0. df = 0.99.

EDITRAT is a flag to set to write out net rates for plotting. df = 0.

EDOTRMIN is used to stop the specification of the zone temperature when the zone heat release exceeds EDOTRMIN.

EFF(MXI*MXKE) Third body efficiencies for each specie in selected reactions. Use together with LISTRE. The first IMX values in EFF refer to LISTRE(1), the next IMX values for LISTRE(2), and so on. df = 1.0

MXKE (maximum number of reactions in which the efficiency of species can be changed) is 10.

ERLIM If $R_k > ERLIM$ times $Rmax$, “*” is printed instead of “=” in zone rate edit. Allows one to tell if reaction is too large. See Eq. (7.4). df = 0.5.

ETA1 η of Eq. (5.6). df = 1.0.

ETA2 Fraction of zone width dynamic rezoner is allowed to move boundary through in one cycle. df = 0.2.

ETA3 Fraction of quantity in zone allowed to be removed from zone by second order remap before remap drops to first order. df = 0.4.

EULER Flag for Eulerian calculation. Eulerian calculation done if nonzero. df = 0.0.

EULER2 If equal 0.0, first order remap of Eq. (5.3) is done. If equal 1.0, second-order remap of Eq. (5.5) is done. If equal -1.0, second-order remap of Eq. (5.7) is done. For laminar flames set equal -1.0. df = 0.0.

FIX is used with TFIX(J) to prescribe zone dependent heat transfer. FIX is the reciprocal time constant for heat transfer. Set it to a large (e.g. 1.e10) to absolutely hold the temperatures fixed at TFIX(J). df = 0.

FNERTIA Factor multiplying left side of Eq. (2.9). If small, damps out sound waves, since it cuts down inertial resistance to pressure gradients. Never set to zero. $df = 1.0$.

FRC(K) K = 1,KMX A_k of Eq. (2.2) for forward reaction rate. Usually read from CDAT but can be modified in input. Modification must be nonzero since code keeps CDAT value if input is zero. If real zero is desired, input a very small number. K must refer to index in problem, not index in CDAT. If NREACT is nonzero, special reactions with the form of Eq. (7.9) will have indices in the problem that are less than the indices of the normal reactions. See NREACT.

FRCMUL(MXK) are multipliers of A-factor for forward reaction rates. $df = 1.0$

FRP(K) K = 1,KMX b_k of Eq. (2.2) for forward rate. See FRC.

FRX(K) K = 1,KMX E_k^a of Eq. (2.2) for forward rate. See FRC.

GAUSDY See C2, S2. Must be nonzero if C2 or S2 nonzero and dynamic rezoner is used. $df = 0.0$.

GAUSX See C2, S2. $df = 0.0$.

H0(I) I = 1,IMX Enthalpy at 298.15° K in *kcal/mole*. See CPA1.

IBNDRYL is a flag for the left boundary condition.

= 1 means fixed wall.

= 2 means open boundary with prescribed pressure PRBL.

= 3 means prescribed boundary flux (See CBURNR).

$df = 0$. If = 0, SETUP will reset it to 3, if PRBL(1) is greater than 0.; else it is set to 1.

IBNDRYR is a flag for the right boundary condition.

= 1 means fixed right boundary.

= 2 means open right boundary with three options: pressure lookup of pressure and time, table lookup of pressure only, or pressure specified by a cubic.

$df = 0$. If = 0, SETUP will reset it to 2 if PRBR(1) is greater than 0. or if NPRBRENG is greater than 0; else it is set to 1.

ICDAT(MXI) is a flag to use NASA coefficients instead of CDAT data.
Set ICDAT(I) = 1 for specie with index I.

ICMXA, **ICMXB** are 8 character strings. df = ". If set, species last with names whose first 8 characters match the strings will be involved in the ICMXEDIT.

ICMXEDIT is a flag for a special edit. df = 0 .

= -1 means the maximum of specie with name ICMXA * specie with name ICMXB.

= 1 means edit max of specie with index ICMXEDIT.

= -1 means produce a zone edit at the time that the product of CO and O concentrations reaches a local maximum. Used to simulate shock tube experiments which use CO₂ chemiluminescence as a diagnostic.

= I, where I is greater than 0 means produce a zone edit at the time when the concentration of species I reaches a local maximum in time.

IEDIT4 is a flag to list reactions rates in EDIT4 as fractions of the total rate (if = 0) or as fractions of the total chemistry rate (if nonzero). df = 0 .

IEDOTR If set to a positive value, compute and edit the location and value of the point of maximum chemical heat release. Generally used in laminar flame problems to locate the flame position. df = 0 .

IFLAME is the same as IEDOTR, but here the flame position is defined as the location where interpolated zone temperature is exactly equal to TFLAME(1) or TFLAME(2). df = 0 .

IFPISTON is a flag for the free piston model. Set to 1 to use the model. df = 0. See also CMPRATIO.

IFUELCS is the specie for the fuel consumption edit. df = 0 for no edit. (See IPRINT(7).)

IHEDIT If positive, replace VSOUND with HEDIT in EDIT1. df = 0 .

IHTREL is a flag to include wall heat transfer only until zone 1 shows positive rate of heat release. Used in thermal ignition problems. df = 0 .

ILAVID If nonzero, it is the index of the reaction whose forward rate will be multiplied by XLAVID, when the temperature falls in the range [TLAV1,TLAV2]. df = 0 .

IPMXEDIT If pistive, edit the maximum rate of pressure rise and the time when it occurs. df = 0 .

IPRINT(L) **L = 1,9** Number of cycles between call to the possible problem edits. IPRINT(L) = 0 means edit number L is not called by this sentinel. See also LCYC(L), MCYC(L). The possible edits are:

IPRINT(1) General edit of problem designed to give overall picture of what is happening in problem. df = 100.

IPRINT(2) Debug edit giving behavior of iterations. Will print out how variables change each iteration during cycle. See also LONGE2. df = 0.

IPRINT(3) Causes problem dumps to be taken. Graphics and restarts are made from these dumps and taking them is generally very cheap. May want to increase default frequency by an order of magnitude. The dumps are written into files named DFILE0, DFILE1, etc. See also MAXDUMP. df = 100.

IPRINT(4) Zone rate edit. Gives printout of all rates affecting variable changes in zone. See also EDIT4MN, EDIT4MX, LSTE4. df = 0.

IPRINT(5) Gives edit of species mole fractions for problem. df = 100.

IPRINT(6) Edit of Lagrange and Eulerian (if appropriate) relative changes of physical variables per cycle. df = 0.

IPRINT(7) Fuel consumption edit. df = 0 .

IPRINT(8) Edit cycle, time, time step, and temp of zone 1 to the terminal. df = 0 .

IPRINT(9) Edit time, mass of certain species and total moles to a separate ASCII file named 'rad'. df = 0 .

IRF(K) **K = 1,KMX** For reactions of the form of Eq. (7.1), this is the reaction type according to the kinetics calculation description, and is never changed. For special reactions of the form of Eq. (7.9), a nonzero value acts as a flag to include c_{tot} in the rate. df = 0.

IRCYMX Maximum number of times the code will cut timestep and retry cycle before quitting when iterations are not converging in ITERMX tries. df = 5.

IRS(K,L) **K = 1,KMX**; **L = 1,4** Index, in CDAT file, of species in reaction K, according to definition outlined in kinetics calculation description. Never changed for regular reactions, but must be set for special reactions with form of Eq. (7.9). See NREACT.

ISANDTR Flag to turn on Sandia transport package. df = 0 for not on. (See DMUL, NAMLINK, and NTRANFRQ.)

ISP(K,L) **K = 1,KMXS**; **L = 1,3** Index, in CDAT file, of c_m in special reactions with form of Eq. (7.9). See NREACT.

ITERMX Maximum number of iterations code will make in trying to solve nonlinear equations of motion before cutting timestep. df = 20.

ITIMER Sentinel for timing select subroutines. Set nonzero if timing information desired. df = 0.

ITZ1MX Maximum number of iterations on startup cycle to solve for α of Eq. (6.2) so that Eq. (6.5) is satisfied. Normally not changed. df = 20.

ITZ2MX Maximum number of iterations of w to solve Eq. (6.7). If HCT reaches this maximum when generating a new problem, might want to increase over default. Default should be adequate for a running problem. df = 20.

IWALL flag to set WALL in CALBDY. df = 0 .

$$\begin{aligned} \text{WALL} &= \text{WALL1}(\text{IT3}) + (\text{WALL1}(\text{IT3+1}) - \text{WALL}(\text{IT3})) \\ &\quad * (\text{TIME} - \text{WALLT}(\text{IT3})) / (\text{WALLT}(\text{IT3+1}) - \text{WALLT}(\text{IT3})) \end{aligned}$$

IZEDIT If positive, add forward and reverse reaction rates, temperature, and species concentrations for zone IZEDIT to usual kinetics edit. df = 0 .

KEQ(MXK) For each reaction if KEQ(K)

= 1, use forward rate and equilibrium data to compute reverse rates;

= -1, use reverse rate and equilibrium data to compute forward rate;

= 0, use both rates from CDAT.

= 2, use the CHEMKIN method for re-calculating reverse rates.

= -2, use the CHEMKIN method for re-calculating forward rates.

df = 0 .

LCYC(L) **L = 1,6** Turns on edit L for every cycle .GE. LCYC(L). df = MXCYCLE + 2.

LIST24(MXI), **LIST58(MXI)**, **LISTMETH**, **LISTNOCT** are the indices (problem-based) of species which are C2-C4, C5-C8, methane, and n-octane respectively. These indices are used to compute the mass of all C2-C4 species, C5-C8 species, methane, and n-octane and the total moles for EDIT9. (See IPRINT(9).)

LISTR(K) **K = 1,KMX** Index in CDAT file of reaction to be used. df = 0. If any LISTR(K) is negative, the reaction name is set to “NO REACTION”, and the rates are set to 0.

LISTRE(MXKE) is list of reactions (indices of reactions in input LISTR, rather than CDAT numbers) for which variable third body efficiencies are to be used in rate expression. Use with EFF (Q.V.).

LISTRF(MXKF) is list of reactions (values in problem input) which are treated as “unimolecular” reactions in the fall-off region whose rate constants conform to the Lindemann form,

$$k = \frac{k^\infty}{1 + \frac{\alpha}{[M]}}$$

where $\alpha = ARC * T^{ARP} * \exp(-ARX/RT)$. The parameters for k^∞ are specified in the usual manner and given in the main reaction listing for a problem.

LISTRLIN(MXKF) is a list of reactions numbers whose rates are treated as “unimolecular” reactions in the falloff regime. The reaction numbers correspond to the order listed in the inputfile. Their rate coefficients conform to the form,

$$k = k_\infty \left(\frac{P_r}{1 + P_r} \right) F$$

where the reduced pressure P_r is given by

$$P_r = \frac{k_0[M]}{k_\infty}$$

$F = 1$, for the Lindeman form used in this case.

$$k_0 = RCLIN * T^{RPLIN} * \exp(-RXLIN/RT)$$

M = total concentration, CTOT.

The parameters for k are specified in the usual manner and given in the main reaction listing for a problem. The above reaction rate specification was chosen to correspond closely with that given in CHEMKIN II.

Related variables include: RCLIN(MXKF), RPLIN(MXKF), RXLIN(MXKF), where MXKF = 10.

LISTRTRO(MXKF) is a list of reactions numbers whose rates are treated as “unimolecular” reactions in the falloff regime. The reaction numbers correspond to the order listed in the inputfile. If the reaction being specified in the forward direction, LISTRTRO is positive. If the reaction is being specified in the reverse direction, LISTRTRO is negative. See LISTRLIN for the assumed form of the rate coefficient.

$$k_0 = RCTRO * T^{RPTRO} * \exp(-RXTRO/RT)$$

F is given by the Troe form:

$$\log F = \left(1 + \left(\frac{\log P_r + c}{n - d(\log P_r + c)} \right)^2 \right)^{-1} \log F_{cent}$$

where

$$\begin{aligned} c &= -0.4 - 0.67 * \log F_{cent} \\ n &= 0.75 - 1.27 * \log F_{cent} \\ d &= 0.14 \end{aligned}$$

$$\begin{aligned} F_{cent} &= (1 - ATRO) \exp(-T/T3TRO) \\ &\quad + ATRO * \exp(-T/T1TRO) + \exp(-T2TRO/T) \end{aligned}$$

The last term is often not used, so the default for T2TRO is 1.0e100. The defaults for ATRO, T3TRO, T1TRO are zero. The above reaction rate specification was chosen to correspond closely with that given in CHEMKIN II.

Related variables include: ATRO(MXKF), RCTRO(MXKF), RPTRO(MXKF), RXTRO(MXKF), T2TRO(MXKF), T3TRO(MXKF), where MXKF = 10.

LISTRSRI(MXKF) is a list of reactions numbers whose rates are treated as “unimolecular” reactions in the falloff regime. The reaction numbers correspond to the order listed in the inputfile. See LISTRLIN for the assumed form of the rate coefficient.

$$k_0 = RCSRI * T^{RPSRI} * \exp(-RXSRI/RT)$$

F is given by the SRI form:

$$F = \left(ASRI * \exp\left(-\frac{BSRI}{T}\right) + \exp\left(-\frac{T}{CSRI}\right) \right)^x dT^{ESRI}$$

where

$$X = \frac{1}{1 + \log^2 P_r}$$

The above reaction rate specification was chosen to correspond closely with that given in CHEMKIN II.

Related variables include: ASRI(MXKF), BSRI(MXKF), CSRI(MXKF), DSRI(MXKF), ESRI(MXKF), RCSRI(MXKF), RPSRI(MXKF), RXSR(MXKF), where MXKF = 10.

LISTS(I) I = 1,IMX Index in CDAT of species to be used. Must specify at least one species. df = 0.

LISTTVR(MXTVR) is a list of indices of those reactions that have time-varying rates, prescribed by:

$$FRC(LISTTVR(K)) = 1.064 * C1TVR(K) * \exp\left(-2.773 * \left(\frac{C2TVR(K) - TIME}{C3TVR(K)}\right)^2\right)$$

Added for a nuclear chemistry application of HCT.

LONGE2 If nonzero, IPRINT(2) gives edit of every variable's change during iteration. If zero, only those variables that have not converged are printed. df = 0.

LPLOT1Z is a flag for edit for 1-zone comparison plots. df = 0 .

LSTE4(J) J = 1,JMXM2 List of zones for which IPRINT(4) will give zone rate edit. Do not count dummy boundary zones in calculating zone numbers. df = 0.

MASSAVG If set to 1, the code will calculate and edit mass-averaged, total problem PBAR and TBAR. df = 0 .

MAXDUMP Maximum number of cycle dumps written into single dump file by IPRINT(3). df = 100.

MBOXID(3) is the box (**MBOXID(1)** = 'box ann') and problem id (**MBOXID(2)** and **MBOXID(3)**). (Total of 24 characters)

MCYC(L) L = 1,6 Turns on edit L for each cycle .LE. MCYC(L). df = 1 if IPRINT(L) .NE. 0; df = 0 if IPRINT(L) .EQ. 0 (default is to print edit L on first cycle if edit called during rest of problem).

MXCYCLE Maximum number of cycles problem will run before quitting. Be sure to set. df = 0.

MXMZ Maximum number of steps in ξ used to integrate Eq. (6.17). df = 100.

MXSKPJ Maximum number of iterations problem will go before updating T^m in Eq. (4.8). df = 20.

NAMDUMP Name of dump file for restarts. When reading from NAMDUMP, the code picks up the cycle number, time, timestep and the values of $c_{i,j}$, T_j , Δr_j and $v_{j+\frac{1}{2}}$ for a particular cycle. The rest of the parameters are set as usual from the input deck. See also NDUMP0 and NCYC0. df = 4HNONE (code assumes not restarting a problem).

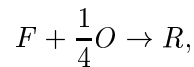
NAMLINK Name of CHEMKIN link file. df = 'tplink'.

NCYC0 In restarts, code will obtain initial values of the physical variables at cycle NCYC0 from NAMDUMP. If NCYC0 = -1, problem will restart from last dump in restart file. Also see NAMDUMP. df = -1.

NDUMP0 When restarting from one of a problem's dumps in NAMDUMP, NDUMP0 may be used to specify the number of the restart dump in NAMDUMP. This is not the cycle number of the dump, but rather the sequence number in NAMDUMP. Usually requires knowledge of which dumps correspond to which problem cycle. This information is printed in HSP output. Also see NAMDUMP. $df = 0$.

NREACT Number of special reactions of the form of Eq. (7.9). Carried internally in code as KMXS. See IRF, IRS, XNP, ISP, XLP, FRC, FRP, FRX, RNLHS and RNRHS. Only one reaction is specified for each special reaction entry, i.e. no reverse rate is implied.

For example, suppose one wants the nonstandard overall reaction



where F represents fuel, O represents oxidizer and R represents an intermediate radical. Suppose we are using species 16, 4, and 94 in the CDAT file to represent F , O and R . Furthermore, assume Eq. (7.9) is to give

$$R = f(T)c_{16}^{\frac{1}{4}}c_4^{\frac{3}{4}}c_{94}^1.$$

with nonzero FRC, FRX in the $f(T)$ of Eq. (7.2). The following input will set the appropriate data with $K = 1$:

```
((IRS(L,M),L=1,1),M=1,3)=16 4 94
((XNP(L,M),L=1,1),M=1,3)=1. .25 -1.
((ISP(L,M),L=1,1),M=1,3)=16 4 94
((XLP(L,M),L=1,1),M=1,3)=.25 .75 1.
FRC(1)=1.2E14
FRX(1)=2.9E4
RNLHS(1)=8H F+1/4O
RNRHS(1)=5H R
```

Note how the two-dimensional arrays IRS, XNP, ISP, and XLP are read in.

The special reactions will have problem indices less than the indices of the reactions read from CDAT. The order in the code is special reactions first, followed by the regular reactions as encountered in LISTR. See FRC.

NTRANFRQ Frequency of call to TRANSCOF (interface to Sandia transport package). $df = 1$. (See ISANDTR, DMUL, NAMLINK.)

NZONES Number of physical zones. Carried internally as JMXM2. Must be set. $df = 0$.

NZXCESS N^* of Eq. (6.5). Be sure to set if dynamic rezoning is done. $df = 0$.

PISTON Volume at top dead center. Usually set to 1.e-4. Used when zone volume is specified by slider crank angle formula. See RPM.

POLYTROP is an option to CALBDY. If greater than 0.0, CALBDY will the polytropic coefficient to compute the volume (at the fixed left boundary). $df=0.0$.

PR0 Initial pressure in CGS units. If set nonzero, code will scale all $c_{i,j}$ so that there is constant pressure PR0 in the problem. Especially useful if species distribution is nonuniform. $df = 0.0$.

PRBL Pressure in CGS units at left ($x = 0.$) boundary. Setting this variable nonzero will result in left boundary being open and allow material to flow in and out of it. If zero, left boundary is fixed wall with no transport through it allowed. $df = 0.0$.

PRBMUL is a scaler for input pressure histories. This provides a convenient means of scaling a large array of prescribed pressure data. $df = 1$.

PRBR Pressure at right ($x = XMAX$) boundary. See PRBL. $df = 0.0$.

PRBRENG(MXPRBRE) Used to specify the time variation of pressure at the right boundary as a table of pressure and time entries. PRBRENG is an array which gives the pressure values in the table. TIMDAT is the array which gives the corresponding time values. A linear interpolation is performed for times between table entries.

PREDIT2 Factor multiplying third term on right side of Eq. (4.14), which give extrapolated variables at end of cycle. Makes possible use of only first-order extrapolation. Usually not set. $df = 1.0$.

PRL1(MXP3), PRL2(MXP3), PRL3(MXP3) Left boundary pressure can be prescribed by a cubic expression in time,

$$pressure_{left} = PRBL + PRL1 * t + PRL2 * t^2 + PRL3 * t^3$$

A different cubic expression can then be used in a subsequent time interval. PRL1(1) is for the first time interval, PRL1(2) is for the second time interval, etc. Time intervals are prescribed with TSTARTP3. Constant boundary pressure is a special case, with PRBL(1) = desired value, all others, including PRL1, PRL2, and PRL3 = 0.

MXP3 (maximum number of pressure cubics) is 6.

PRR1(MXP3), PRR2(MXP3), PRR3(MXP3) Same as PRL1, PRL2, and PRL3, but for right boundary.

PRMUL If zero code bypasses hydro calculation. If PRMUL = 0.0, keep QK = 0.0. df = 1.0.

QK Viscosity coefficient λ of Eq. (2.12). df = 0.0.

RDELCMX Maximum relative change per cycle allowed physical variables except species whose mole fraction is less than CNEGF. df = 0.05.

RDELCNEG Maximum relative change per cycle allowed species whose mole fraction is less than CNEGF. df = 0.5.

RDELW1 Maximum relative change allowed in w when updating the α of Eq. (6.2) to satisfy Eq. (6.5). df = 0.2.

RDELW2 Maximum relative change allowed in w when integrating Eq. (6.17). df = 0.05.

RDOFF Number on the order of machine roundoff. Used to keep divide-by-zero's from occurring. df = 1.E-14.

RELDC(I) I = 1,IMX Used to scale D_i^0 . See DCZERO. df = 1.0.

RHOZERO Used in calculation of XCH. Set for zone 1 (zone mass) in SETUP.

RLIM k_1 of Eqs. (7.5) and (7.6). Used to define R_{max} . Users should check that default is adequate for their problems. Also see TAU LIM. df = 1.E16.

RNAMES(LS,MXI) Names of reactant in Hollerith or BCD form. Each name is LS Cray words long. (Currently, LS = 2, so the names are 16 characters long.) Used when desired species are not available in CDAT or when they must be changed for some reason. df = taken from CDAT based on LISTS.

RNLHS(LR,MXK) Hollerith description of left side of reaction equation. (Currently, LR = 4, so a side of the reaction name is at most 32 characters.) Used in edits. Usually taken from CDAT file except for special reactions of form Eq. (7.9). Should be centered in machine word. See NREACT for example.

RNRHS(LR,MXK) Hollerith description of right side of reaction equation. See RNLHS.

RPM One of the parameters used to allow the zone volume to change with time in the same manner as the combustion chamber volume in a reciprocating engine. The zone volume is specified as a function of time according to the slider crank angle formula.

$$\text{zone volume} = PISTON - \frac{5.236 * 10^{-6} RPM (COMPRESS - 1) [\sin(CA) + 0.5 STROKE \sin(2CA)]}{\sqrt{1 - (STROKE)^2 \sin^2(CA)}}$$

where CA=0.10472*RPM*TIME-TT, CA is 0 at top dead center, RPM is the engine speed in revolutions per minute, COMPRESS is the compression ratio, and STROKE is the half stroke to rod ratio.

RRC(K) K = 1,KMX A_k of Eq. (2.2) for reverse reaction rates. See FRC.

RRCMUL(MXK) are multipliers of A-factor for reverse reaction rates. df = 1.

RRP(K) K = 1,KMX b_k of Eq. (2.2) for reverse reaction rates. See FRC.

RRX(K) K = 1,KMX E_k^a of Eq. (2.2) for reverse reaction rates. See FRC.

RZCNVG Convergence criteria on solution of Eq. (6.7) for w . The code iterates until w changes less than RZCNVG. df = 0.01.

RZERO is sum of time step * velocity in zone 1. df = 0.

RZFLAG Flag for dynamic rezoner. Must be set nonzero if dynamic rezoning desired. See also DELYMN. df = 0.0.

S0 Used to initialize nonuniform zoning to match step function. Works similarly to C0(I) but physical variables must be set independently. Useful if variables are initially constant but one wants to choose initial zoning so that resolution present when source turned on. As with C0, (see also Eq. (9.3), let

$$\alpha(x) = \frac{1}{1 + \exp\left(\frac{x - XDISC}{XSPRD}\right)}$$

The code will initialize zoning to describe

$$S(x) = \alpha(x)S0 + [1 - \alpha(x)] S1 + \beta(x)S2.$$

For description of $\beta(x)$, see S2. NZXCESS and XSPRD should be nonzero. Note that since S0 and C0 use the same variables to define α , one cannot use both in same problem.

S1 See S0.

S2 With GAUSX and GAUSDX, allows one to initialize zoning to match a localized distribution characterized by the function

$$\beta(x) = \frac{4}{[1 + \exp(-y)][1 + \exp(y)]}$$

where

$$y = \frac{x - GAUSX}{GAUSDX}$$

$\beta(x)$ has a maximum of 1 at GAUSX, and falls exponentially to zero over a distance characterized by GAUSDX. We use this form rather than a true Gaussian because it is the derivative of Eq. (9.3), so that the indefinite integral analogous to Eq. (9.4) is easily done. See S0.

SORPCC S^{vol} of Eq. (9.1). See also SORSPRD, SORXMAX, SORSWF and TIG. df = 0.0.

SORPGM S^{mass} of Eq. (9.2). See also SORSPRD, SORXMAX, SORSWF and TIG. df = 0.0.

SORSPRD Δx of Eq. (9.3). Must be nonzero if used. See also SORPCC, SORPGM. df = 0.0.

SORSWF When nonzero, code checks to see if rate of energy production due to reactions is greater than SORSWF times source energy rate. If it is, sources are turned off. Provides a way of cutting off sources when reactions have started. See also SORPCC, SORPGM. df = 0.0.

SORXMAX x_0 of Eq. (9.3). See also SORPCC, SORPGM. df = 0.0.

STROKE Stroke of engine. Used when zone volume is specified by slider crank angle formula. See RPM.

TAUILIM $1/\tau$ of Eq. (7.7). Used to define R_{max} . User should check that default is adequate for their problems. See also RLIM. df = 1.E20.

TEMPMX Maximum temperature. If non-zero, code execution stops when temperature in zone 1 exceeds TEMPMX. df = 0.0.

TENG(MXPRBRE) An array used to specify the time variation of TWALLX by interpolating from a table of values. (See WALL.) The temperatures in the table are specified by TENG. The corresponding times in the table are specified by the array TIMDAT. A linear interpolation between table entries is performed. df = 0. MXPRBRE (maximum number of table pressure values) = 400.

TFIX(MXJM1) TFIX(J) is the wall temperatures for zone J. Used with FIX(Q.V.) to include heat transfer to all zones of a problem. df = 0.

TFLAME(2) are two temperature values which can be used to define flame location in a multizone problem. Exact locations are linearly interpolated between zone center values. df = 1500., 2000.

TIG Maximum time of energy deposition. Sources are on for times less than TIG. See also SORPCC, SORPGM. df = 0.0.

TIMDAT(MXPRBRE) An array of times used to specify the time variation of TWALLX as a table. See WALL and TENG. Also used to specify the time variation of pressure at the right boundary. (See PRBRENG.)

TIME is the problem time variable. It can be set initially to any value, but is usually 0. initially.

TIMEMX Maximum problem time. Must be set to run past first cycle.
df = 0.0.

TIMREF is used to offset the time a table lookup of pressure or temperature is performed.

TIMRPM is used to modify the time that a table lookup of pressure or temperature is performed, so that the same pressure or temperature history can be used for different engine speeds.

TLAV1, TLAV2 select a temperature window ($TLAV1 \leq TEMP \leq TLAV2$) within which to modify a give reaction rate. See ILAVID and XLAVID. Defaults: $TLAV1 = 0.$ and $TLAV2 = 1.e10.$

TSTARTP3(MXP3) is an array used when specifying the time variation of pressure of the right-hand boundary as a series of cubics. MXP3 (maximum number of pressure cubics) is 6.

TSTOPT3(MXT3) See TWALL(I). MXT3 (maximum number of temperature cubics) is 25.

TWALL(MXT3), TWALL1(MXT3), TWALL2(MXT3), TWALL3(MXT3)
Used to specify TWALLX for the case of a constant wall temperature.
The time dependence of TWALLX is specified as a series of cubics.

$$TWALLX = TWALL(I) + TWALL1(I)*t + TWALL2(I)*t^2 + TWALL3(I)*t^3$$

for $TSTOPT3(I-1) \leq t \leq TSTOPT3(I)$ where t is time. Note that the first cubic extends from $0 \leq t \leq TSTOPT3(1).$

WALL A parameter used to allow heat transfer out of the left boundary. The heat loss is proportional to $WALL*(T-TWALLX)$, where T is the temperature of the zone adjacent to the left boundary. TWALLX is specified as a constant temperature, TWALL, or is time dependent. The time dependence of TWALL can be specified as a series of cubics (see TWALL(I)) or as a table of values (see TENG).

WALL1(MXWALL) See IWALL. MXWALL = 25.

WALLT(MXWALL) See IWALL.

XDISC See C0, S0.0. df = 0.0.

XFLSTOP is position to stop execution when either flame reaches it. $df = 1.e+20$.

XKC(MXK), **XKP(MXK)**, **XXK(MXK)** specify specie equilibrium constants. Values are read in from CDAT, but user may override them in input.

XLAVID is multiplier of forward rate (for LAVID option). $df = 0$.

XLP(K,L) **K = 1,KMXS**; **L = 1,3** The exponential powers α_m in Eq. (7.9). Must be set for special reactions. See NREACT. $df = 0.0$.

XMAX Maximum coordinate in problem. Left boundary of problem is always zero. Must be set. $df = 0.0$.

XNP(K,L) **K = 1,KMXS**; **L = 1,4** Stoichiometric coefficients for species given by IRS array for special reactions of the form of Eq. (7.9). A positive number β means β molecules are destroyed in reaction. A negative number β means β molecules are created by reaction. Must be set for special reactions. See NREACT. $df = 0.0$.

XSPRD See C0, S0. Must be nonzero if C0 not equal to C1 and dynamic rezoner is used. $df = 0.0$.